Evaluation Framework for Technology Agnostic Hybrid NoC Architecture

A thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Technology

by

Raghav Kishore

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

May 2016

To my parents.

Certificate

This is to certify that the thesis titled "Evaluation Framework for Technology Agnostic Hybrid NoC Architecture" being submitted by Raghav Kishore (MT14067) to the Indraprastha Institute of Information Technology Delhi, for the award of the degree of Master of Technology, is an original research work carried out by him under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree or diploma.

May, 2016

Dr. Sujay Deb
Department of Electronics and Communication Engineering
Indraprastha Institute of Information Technology Delhi
New Delhi 110 020

Acknowledgements

I would like to express deepest gratitude towards my advisor Dr. Sujay Deb and thank him for giving me the opportunity to work with him on challenging research problems in Network-on-Chip. The insights into the semiconductor industry together with the trends in exa-scale computing really motivated me to pursue this work. I thoroughly enjoyed working with him and his research group-Advanced Multicore Systems Lab. I would like to sincerely thank Hemanta Mondal for his constant support, for using my work for his PhD research and for the endless brainstorming sessions that we use to have. I really appreciate the way he used to present problems in performance evaluation of new topology which used to excite me further to find a solution to. I would like to also thank Harsha Gade for his support and help during the development of this framework. It will be incomplete without acknowledging the useful technical discussions which I had with Praveen Kumar and Wazir Singh regarding various aspects of multicore technology. My stint as an intern during this phase with CPU-GPU Performance Analysis and Benchmarking team at ST Microelectronics in Noida, equipped me with the processor core performance perspective thereby complementing my understanding and also aiding my work here at the network level. At last I thank my parents for their constant support throughout this journey.

Abstract

Performance of emerging multicore systems is significantly influenced by the efficiency of the communication fabric that glues the entire system together. Large number of cores results in excessive energy consumption while also increasing hop count for far apart nodes. This severely degrades network latency and throughput thereby resulting in performance bottleneck while using conventional planar metal interconnects. However, advancements in emerging network-on-chip (NoC) interconnect technology opens up possibilities to explore energy-efficient high-performance but unconventional and application-specific topologies. In this thesis, an evaluation framework for technology agnostic hybrid NoC architecture is presented. Challenges in regular NoC design space exploration are highlighted emphasizing on the increased degree of complexity in case of hybrid NoCs. Generic simulator modification guidelines are also included in view of no available standard procedure to modify existing simulators for hybrid topology. Router utilization based metric is introduced and is applied to enable design and evaluation of energy-efficient architecture by implementing dynamic voltage scaling (DVS) with power-gating. Extension to simulated annealing based floorplan engine is also proposed which not only makes the flow floorplan-aware but it also demonstrates the extensibility and agility of the framework for further development. The proposed framework is implemented on top of an existing cycle-accurate open-source network simulator. Detailed performance analysis and benchmarking undertaken in four different case studies demonstrates the cost-effectiveness and efficiency of this framework. Hybrid topologies proposed using this framework are energy-efficient and high performance.

List of Figures

List of Tables

Contents

1. Introduction

Modern system-on-chips (SoCs) pack diversified set of elements like CPUs, GPUs, DSPs, memory, etc. together on a single chip making them not only computationally powerful but also very complex at the same time. These high performance chips are more general in nature to be able to meet many application needs. The heterogeneity on the system enables efficient use of the existing available resources instead of opting for entirely new hardware when there is a shift in application requirements. This surge in computing is driven by new applications such as autonomous driving, server and data center, internet-of-things (IoT) which involve computationally intensive and parallelizable tasks like machine learning, big data analytics, computer vision, artificial intelligence, etc. IBM recently released a sixteen-core single-board brain-inspired chip with ultra-low-energy consumption and backs its ability to scale up to meet the demands in exa-scale computing without consuming enormous amounts of energy [1]. While IBM moves towards neuromorphic chips, AMD on the other hand is moving towards heterogeneous computing and describes it as the next big thing in computer architecture [2][3]. Clearly, across industry, there is an increasing trend towards multicore chips to meet the performance demands of modern and futuristic applications.

The on-chip intra-core communication fabric is an important element towards realization of these multicore systems. The interconnect fabric serves as the backbone of the entire chip and acts as a glue to hold all the cores together. The scalability is tremendous, but brings with it an energy and area overhead which can be significant percentage of overall chip energy and area. As the systems scale, the network performance also gets affected. Issues of higher packet latencies, lower router throughputs, increased network congestion and chip hotspots make it very challenging to meet the desired network performance in order to be able to meet overall multicore chip performance targets. This led to experimentation focused solely on the network performance which is the bottleneck in multicore system design [4]. In order to improve packet latencies and throughput, it is suggested to augment the conventional networks-on-chip

(NoCs) mesh topology with single-hop long-range and high-bandwidth links between selected far apart router nodes [5]. For larger systems it is suggested to opt for hierarchical topology with different sub-nets [6]. Though the performance gain by moving from homogeneous NoCs to heterogeneous NoCs is very promising, the main drawback in the physical realization of such networks is limited by the implementation issues experienced in use of long wires in the deep-sub micron (DSM) era. As the wire delay grows exponentially with the wire length, it is very difficult to meet timing with long-range on-chip wired links. Radical solutions have been proposedl to address this issue. These include, replacing selected wired links with emerging interconnects like optical, wireless, etc [7].

The existing NoC design flow provides little or no support to hybrid network design space exploration. The challenges in the existing network simulation limit the ability to propose efficient network topology for emerging multicore systems.

In this thesis, an evaluation framework for network simulations for emerging multicore systems is presented. Network efficiency is achieved by means of architectural design decisions and the proposed simulated-annealing based floorplanning. The significance of router utilization to achieve the low power heterogeneous NoC is demonstrated. The challenges in existing network simulators are highlighted and also generic modification guidelines to support network heterogeneity included. The multicore system with hybrid NoC is stressed under both real and synthetic benchmarks to evaluate and demonstrate the benefits over regular mesh network topology for same design setup. The results observed emphasize the need of an improved network design framework for emerging multicore systems.

2. Related Work

The first step towards research in computer architecture is selection of a good simulator which provides the basic facilities to study and evaluate the desired architecture. The complexity involved in the emerging multicore systems can only be reflected upon by the available simulators for it. These simulators can be broadly divided into two categories: processor simulators and network simulators. Processor simulators are also known as full-system simulators since they include the ability to model and support a representative set of popular instruction set architectures (ISA) like ARM, MIPS, PowerPC, x86 etc. Most of these simulators boot Linux operating systems and also enable execution of custom compiled tests on the underlying hardware. Support to popular benchmarks like PARSEC, SPLASH, SPEC 2006, etc. is included. These standard benchmarks help evaluate the different architectures to identify performance bottlenecks. GEM5 [8] models CPU and memory sub-system hierarchy and supports large multicore systems on the same network. The verbosity results in drawbacks of very high run times and file size of reports generated making the iterative design process very tedious. Multi2Sim [9] has recently started to gain popularity mainly because of the level of user-configurable heterogeneity built into it. It additionally supports GPUs together with CPUs on the same network. This flexibility is far superior to any simulator currently available. However, mapping of memory elements to same network is limited to only between two levels. This implies that a cache belonging to third level requires a second network connected to cache in second level and cannot directly connect to the first network itself. Though the evaluation reports generated from full-system simulators are highly detailed, they still do not include network level metrics like packet latency, throughput and energy. These metrics are necessary in the design of a good overall multicore system. This shortcoming is overcome by using network simulators.

Standalone network simulators like the cycle accurate Noxim [10] and BookSim [11] are very popular. Advantage is relatively simple source code and fast trace-

based simulations. BookSim is limited by square topology configurations only. NoCTweak [12] provides detailed energy information. The general purpose gpNoCSim simulator is presented in [13] and supports only uniform traffic pattern. NIRGAM [14] is another such tool. To address study issues due to significant energy consumption by communication fabric, recently, some research groups have released energy or power aware network simulators. Ocin-tsim [15] is Dynamic Voltage and Frequency Scaling (DVFS) enabled only while work of [16][17]supports both DVFS and power-gating. NoRd [18] models node-router decoupling for efficient power-gating while work of [19] achieves similar goals but with routing and topology re-configuration. DSENT [20] and Graphite [21] partially support photonic, wireless and wired interconnects.. Graphite runs faster simulations but cuts down on accuracy. Recently released version of Noximsupports hierarchical topology with top layer intended to be completely wireless and also includes a revamped power model. Both these points reflect the acceptance of emerging interconnects in NoC design and the crucial role of network energy in achieving performance needs.

The increasing densities on the chip due to growing number of cores may lead to hotspots thereby severely degrading chip performance. HotSpot [22] tool is intended for architectural studies and used for thermal modeling. ArchFP [23] is a pre-RTL stage floorplan tool intended for chip-multi processors (CMPs) which is intended for early design estimate much before the actual circuit and technology implementation stage. Benefits are seen in the improved IR drop, electromigration and temperature characteristics. Work in [24] describes methodology for wiring efficiency based floor planning during NoC synthesis phase. Whereas work in [25] is about optimization in floorplanning for both electronic and photonic NoCs. From this we can infer that the increasing complexity in the emerging multicore designs, early stage floor plan exploration is beneficial to overall performance of the chip.

The complexity in emerging multicore designs can demand unconventional network topology which is better realized by using advanced interconnects like photonics [26], wireless [27], etc. It is to be noted that multiple aspects like core selection, memory hierarchy, network topology, floorplan, temperature profile, etc. need to be taken care for such systems. The overall design performance requirements can only be met if network characteristic can also be evaluated along with the multicore system topology. An integration of the network simulators with a full system simulator for real application based performance analysis is hence necessary. Topaz [28] and Garnet [29] have already been integrated with Gem5. Sniper [30] supports hundreds of core and enables variability in terms of multi-threading, multi-program workloads and shared memory applications and is integrated on top on Graphite. Similarly, various

plugins are available for various mini tasks, for example, Ruby [8] is intended for memory subsystem linkages while Netrace [31] is a set of C++ libraries intended to address the issue of packet dependency in network simulators for trace-based analysis. However, the tools for which these plugins were originally developed for are now either no longer supported or limited to regular topologies only. Hence, for hybrid (or heterogeneous) topologies, integration between different tools is needed. This requires in-depth understanding of simulator internals and subsequent in-house modification. This is an overhead which is now becoming a necessity and hence makes design and evaluation of such hybrid architectures with emerging interconnects challenging and very time consuming. This calls for the need of a shift in existing network design flow itself to enable quick configurations and evaluations of emerging multicore systems.

3. Emerging Multicore Systems

In this chapter, the emerging multicore systems are presented in detail and emphasis is on the inherent complexity attached in design and evaluation of these systems. The intention is to show the significance of the backbone interconnect network in this new era of computing and its role in achieving overall design coherency and performance.

Accelerated Processing Units (APUs) from AMD are designed to behave as CPU and a GPU on a single chip [32]. These are intended to be fusion processing units without opting for a traditionally different chip for the GPU. Such a design trend clearly indicates higher on-chip integration between the CPUs and the GPUs. Multi2Sim, as introduced previously, permits configuration and subsequent evaluation of both the CPU and as well as the GPU on same system. It is intended for core counts from tens to hundreds and even thousands on the same chip. With its own internal user configurable interconnect, it is one of first tools in open source community available to researchers working on truly heterogeneous computing. Intel recently announced a new Xeon processing unit with integrated field-programmable gate arrays (FPGAs) and boasts of a 20x increase in performance [33]. It allows specific distribution of workloads on the chip and is meant for enterprise and data center applications. Such combination is intended towards workload specific applications wherein with change in workload, the FPGA can be reprogrammed to meet the needs instead of opting for entirely new hardware. Altera suggested combining digital signal processors (DSPs) and FPGA to achieve significantly higher performance at comparatively lower cost [34]. Futuristic trends include integration of CPU, GPU, DSP and FPGA on the same chip. Heterogeneous computing is redefining everyday life with a richer experience. Surge for high definition and ultra-high definition videos with life like quality and the advancements in GPU are indeed making this possible. Vector processing units are advanced forms of GPUs which have multiple computing nodes on single GPU all capable of operating simultaneously to

process large amounts of computation intensive tasks. Futuristic technologies like visual reality and augmented reality all depend on advancements in such heterogeneous computing systems. AMD states that the next era in computer architecture research is of the heterogeneous computing. The Heterogeneous System Architecture (HSA) Foundation, a consortium by major industry leaders and academic institutions is aimed to enable this next generation of computing platforms [3].

EEMBC, a benchmark consortium is working towards workloads of future applications [35]. ScaleMark is intended to meet needs of cloud and big data with ability to deal data in exa-bytes. Such systems are very different from ordinary computing systems. MultiBench is intended to examine multicore architectures and identify bottlenecks and degree of utilization and parallelization. The focus of benchmark developers is to identify and model emerging workloads for future applications and then stress and evaluate the probable hardware intended for it. Performance analysis and benchmarking is now getting increasingly complex due to the exponential growth in complexity of the underlying hardware. Major observations show that the system performance is increasingly getting constrained by the intra-core communication fabric used for given multicore design.

With this paradigm shift from homogeneous towards heterogeneous multicore computing and the increasing role played by interconnects holding this system together, it is of interest to experiment with new and radical network topologies to achieve demanding performance requirements. This is where the role of network simulation comes in and its significance truly exemplified. This change in design philosophy requires newer design flows for network simulation. In the next section we present the challenges in available NoC simulators and also suggest general guidelines in order to suit them to the given purpose.

4. Challenges and Generic Modification Guidelines

In this chapter, major challenges encountered while carrying out in-house modifications in existing network simulators are described. Different research groups have developed and released different network simulators. Due to lack of any generic development guidelines, this has resulted in different NoC models written in different programming languages thereby isolating the tools and making it difficult to compare performance thus evaluated. Intension to support hybrid topologies with long-range links and implementation of energy-efficient techniques only multiplies this complexity multifold. This has resulted as hindrance to NoC research wherein it is required to be able to quickly implement and evaluate a desired topology with ease. Hence, an attempt has been made to outline generic modification guidelines for network simulators to support hybrid topologies.

4.1. Challenges in existing network simulators

4.1.1. Topology

Large numbers of topologies (mesh, torus, folded torus, ring, star, spidergon, hierarchical, etc.) are available but not all simulators provide support all of these. In order to propose a new topology and evaluate it using an existing simulator, significant modifications are necessary. The desired modifications to obtain irregular topologies can be done in two ways; first approach is to make minor changes to the source code of the existing topology and the second approach is by adding a completely new topology to source code. In either case, according to the changes in topology file, corresponding modifications are required in the user's configuration setup and router structure. The traffic pattern and flow control may also require modifications in some cases. Considering a regular 2D mesh topology, generally every router consists of five ports, wherein the boundary line routers also have five ports with some ports left unused. On the

contrary, for irregular or hybrid topologies, additional ports may be introduced in the inland routers that are connected to long-range links. The hybrid topology can be realized by rewiring the existing setup which can lead to a significant improvement in packet throughput and latency.

4.2.1. Routing

Common routing schemes like deterministic XY-routing, adaptive Odd Even (OE), Source routing, are available with most of the NoC simulators. If new topology supports an existing routing algorithm, no changes are required to routing function files. In that case existing routing scheme can be used with the new topology file. Most common routing schemes do not leverage the full potential of wireless links and hence new routing schemes are required for hybrid NoC architectures. For new topology with new routing scheme, modifications are required in routing function and routing table file as well as traffic file. Virtual channel (VC) assignments and intermediate destination choices etc. should be taken care of at the time of introducing new routing schemes. It should be ensured that the modified routing is deadlock and livelock free.

4.3.1. Traffic Pattern

Synthetic traffic patterns like uniform, random, transpose, etc. are commonly available with most of the network simulators. These synthetic traffic patterns might not fully utilize the long-range links and to fully understand the usability of hybrid topologies, different application specific or more practical traffic patterns must be simulated. To simulate different traffic patterns, configuration file needs modification. To add a new traffic pattern, traffic function file needs to be updated properly. The exact definition of new traffic pattern must be added along with other initialization information. The processing unit that will be using that traffic function may also require modifications. All parameters like number of nodes, source and destination pairs, etc. should be defined appropriately in the simulation setup.

4.4.1. Performance Parameters

NoC is generally evaluated by parameters like packet energy, throughput and latency at given network load. These parameters are useful for comparative analysis of multiple competing options and topologies. The validation of newly proposed hybrid NoC can be done by these parameters. But again, not all simulators give complete information about network statistics whereas some do not provide enough details for debug. Mostly, power modeling is carried out on one tool and is either extended or integrated to another tool for network and throughput analysis. Often, there is no insight into actual utilization of network resources during simulation thereby restricting to take architecture design decisions based on real time network activity.

4.2. Recommended modification guidelines for cycle accurate simulators

4.2.1. Understanding the directory structure

Typical source code directory consists of mainly documents, binaries, source files and header file includes. Source directory contain all the topology related files while the binary directories contains the compiled and run time related files for simulator. Includes directory may contain files for pseudo random number generator algorithms, power and energy calculations. Other directories may include files related to compilation, design space exploration and power modeling for the simulator.

4.2.2. Understanding the file structure

It is necessary to understand the hierarchy of all the elements of the NoC modeled in the simulator and this can only be achieved by understanding the file structure. Files related to individual components of the NoC will be located in the source directory. Usually a top level module would contain the NoC and the global parameters. The global parameters file would contain default values which can be modified by the user in the configuration setup file. A parser file may be included which would parse this configuration parameters and pass it to all the sub-components of the simulator. The NoC would contain the repeating tiles and the interconnection fabric. The node would normally contain a processing element and a router including pin mapping for the two sub-components. Processing element would contain the core processing module. Buffer would contain FIFOs for data packet arrival/ departure at each port and might also include virtual channel declaration. The router would link each node with the network.

4.2.3. Understanding the functional flow

Once the file structure is understood, individual functional definitions of the packet, flit, traffic patterns, implementation of transmitting/receiving mechanism at each node, buffer monitoring by routing algorithms, congestion control, low-power link mechanism and selection strategy can then be easily looked into and modified as per need. As an example, addition of a new port to all the routers would include three main steps. First, physical addition of the port, addition of a new direction in the global parameter file and re-wiring of interconnects. Second, required changes in the routing algorithm at each of the routers to be enable use of the newly created hybrid port. Third, corresponding changes need to be made in the parser and user configuration setup files. As the directory and file structure differs from simulator to simulator, additional files may need to be modified as per the simulator under consideration. Hence, the above is just an attempt to present a generalized approach to serve as a guideline to be able to modify any simulator for irregular (or hybrid) NoC architectures.

5. Proposed Evaluation Framework

In this chapter, a step by step modular procedure is adopted to arrive at the proposed evaluation framework for emerging multicore systems with hybrid NoC backbone. This framework facilitates the user to enable these modules as required. All subsequent inputs, as it will be discussed shortly, are user configurable and not constrained to specific values by hard coding. This keeps the framework flexible and generic.

5.1. Multicore System Configuration

The complete emerging multicore system is divided into two parts: processor level and network level. The processor level includes configuration of different computing units while the network level includes configuration of the on-chip interconnect fabric required to hold these complex cores together with other IPs on the system. Processor level configuration is carried out on full-system simulators. GEM5 supports multicore system with popular instruction set architecture (ISA) like ARM, MIPS, x86, etc. and also permits micro-architecture modification. It supports complete cache-memory hierarchy which is a user configurable sub-system using the Ruby framework for custom interconnects. It provides full system emulation as well as system-call emulation capability with multi-threading. Real benchmarks like PARSEC and SPLASH-2 are also supported. Multi2Sim on the other hand supports multicore heterogeneity which permits both CPU and GPU on the same setup. Each of the cores can either be single threaded or multithreaded. Memory sub-system permits custom sharing of cache. For example, each core can be each assigned stand alone L1s while all these L1s might share the same L2. Or the data cache L1 can be assigned to each core while the instruction cache L1 shared between cores. Similarly, different L1s can be grouped together and each group assigned to different L2s and so on. The size of the caches as well as the set associatively can be each custom assigned individually. The caches can even be made to service defined address ranges. Once the appropriate full-system simulator is chosen and multicore setup

whether homogeneous or heterogeneous configured, the next step is to design appropriate network topology to meet required performance requirements. The simulation reports generated from these tools will be used later as packet data traffic trace for design and evaluation of corresponding NoC topology.

5.2. Addition of hybrid port and long-range links

The internal network simulators in full-system simulators offer only bare minimum capabilities and hence a standalone network simulator is needed to design and evaluate NoC properly and completely. In order to support hybrid NoC with long-range links, first step is addition of hybrid port on regular router node Fig. 1.

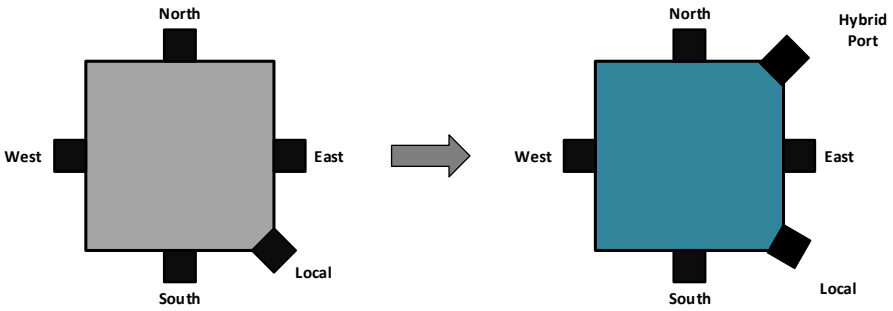


Fig. 1- Addition of hybrid port to regular router node

The purpose of the hybrid port is to connect far apart nodes with single-hop long-range links. Hence, the second step is to create long-range links between desired hybrid router nodes Fig. 2.

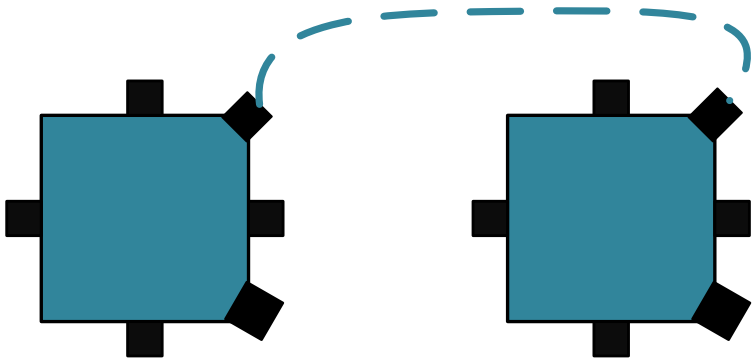


Fig. 2- Long-range link creation

This enables to realize hybrid topology with emerging interconnects wherein, the usual mesh links are wired while the long-range ones realized using wireless, photonics, radio frequency (RF), carbon nano-tube (CNT), etc. This makes the

framework technology-agnostic as it does not differentiate as to how the long-range link is realized. The framework requires energy consumed by the long-range link for corresponding implementation technology only and assumes single-hop communication. The nodes with long-range links are defined as hybrid nodes while the ones without these long-range nodes as regular nodes. In order to create long-range links between desired router nodes, the framework simply reads a user written file which contains information of link pairs. This makes the entire process completely user friendly and re-configurable. It will permit quick changes in topology with ease. With full capabilities of a standalone network simulator, such a provision to create hybrid topology is not supported in currently available network simulators.

5.3. Routing and Arbitration

With the addition of an extra link on each node, corresponding change needs to reflect in routing algorithm as well. Simply linking two nodes will not enable the router to actually route a packet through it. The routing algorithm also needs to be modified to be able to use this extra link effectively. Deadlock and livelock are two major issues which need to be taken care of in presence of an extra link. As per the turn models, packet coming in via this link should not be routed in a manner giving rise to cyclic behavior thereby leading to a deadlock. The work of Small World on-chip networks [5] explains this critical issue using the turn model concept. This model illustrates the turns which are permissible and the ones which are not for routing algorithms under consideration. The basis of turn models is to prevent both deadlock and livelock. Based on the theorems presented and proved in this work, North-Last routing algorithm is recommended for nodes with long-range links, whereas XY routing for all other nodes. The routing information is again user configurable input to the framework. It is indeed interesting to emphasize that how a routing decision inside a router is made is different from how the data packet is actually transferred outside the router. The routing algorithm takes in packet from one input port and assigns it one output port, based on the routing mechanism set. The router on the other hand, transfers this data packet via the output port link. Different link traversal principles might be used depending on the nature of the link. Though this traversal will differ in case of wired, wireless, photonic, etc. but will help to achieve the overall goal of single-hop communications between far apart nodes. In anticipation of congestion at the hybrid router node due to preference of all packets to use the long-range link, corresponding flow control mechanism is inserted which denies use of long-range link in case of heavy traffic. The hybrid (or heterogeneous) routing is illustrated below in Fig. 3.

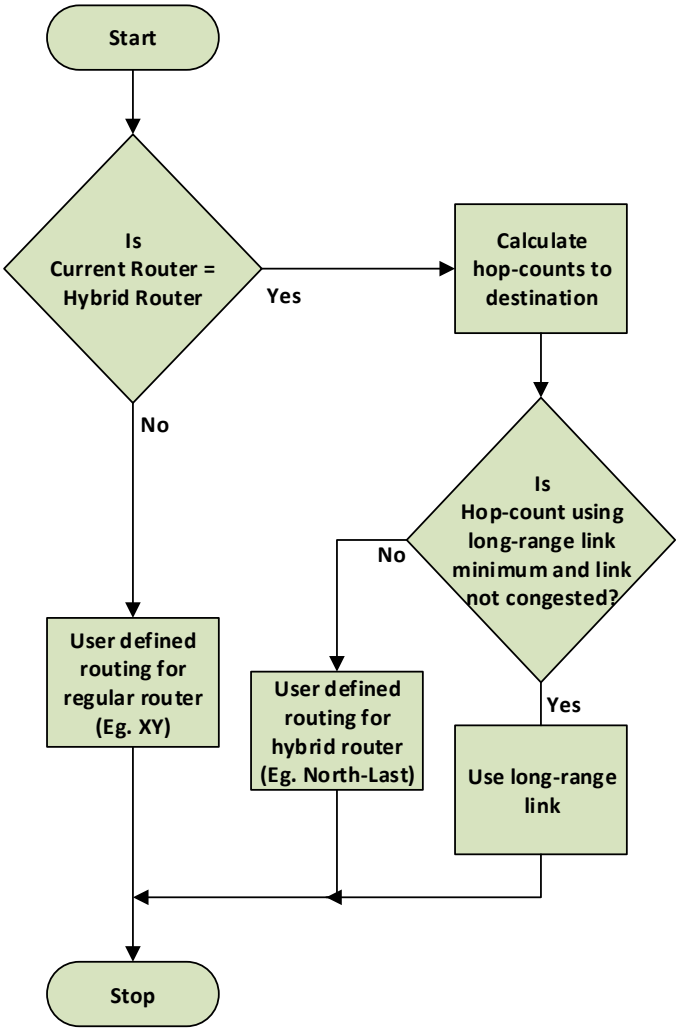


Fig. 3- Hybrid routing algorithm

Inclusion of additional links on hybrid nodes is likely to stress the router arbiter. Packets from one direction may be preferred while packets from other direction might be starved for arbitration. In order to minimize packet starvation, the existing oblivious router arbitration is replaced with round robin router arbitration policy for both the hybrid as well as regular routers. Sanity checks with oblivious arbitration and round robin arbitration for synthetic traffic confirms a higher throughput for the latter. The round robin arbitration algorithm is depicted in the below info-graphic. A pointer is maintained at each router which tracks last served port. In the next clock cycle, this pointer is incremented by one direction. This ensures rotation of starting port for router arbitration (Fig. 4).

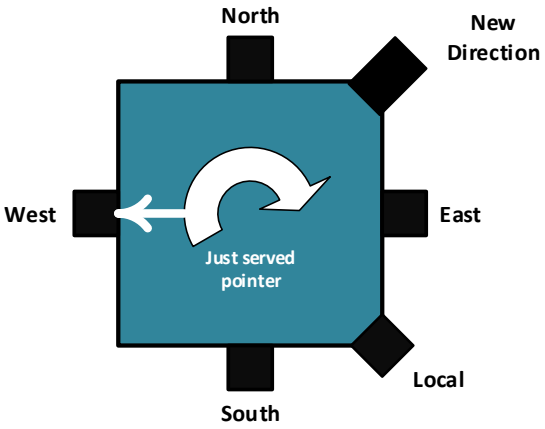


Fig. 4- Round-robin router arbitration

5.4. Utilization Metric

The heterogeneity achieved has its own drawbacks. The addition of extra links and ports and communication mechanisms results in energy overhead when compared to regular architecture. Secondly, with the variability attached to the hybrid nodes that enable the changing of node pairs which are converted into hybrid ones, some nodes and hence links may not be efficiently utilized. Hence, there needs to be some feedback to quantify the real time usefulness of the extra hardware added as per initial estimates. This information can now be used to further optimize and fine tune the topology by adding or removing links, changing node pairs and their placement. To achieve this goal, utilization metric is introduced and implemented for each router node. Of the various possibilities to measure router utilization, per packet routing is adopted. Every router needs to make routing decisions for every incoming packet. The higher the number of packets that visit a router, higher is the number for total routing decisions made. This is reflective of the traffic at that each router is handling at any given time and an abstraction for all other sub-components like buffer occupation status, crossbar traversals, link bandwidth, etc. The router utilization is calculated at every cycle and for each router. A counter is placed inside the router which is incremented every time a routing decision is made. The sum of the counts of each router is added to get total count for that cycle and utilization for a router node is calculated by dividing the count of given router by the total count. Normalization is carried out to represent the utilization of any router in terms of all the routers on the network.

Hence this gives a relative perspective as to which routers are heavily utilized and which are not. Utilization of router, $\mathbf{U_r}$ can be determined as,

$$\mathrm{Ur}_i\Big|_t = \left(\frac{Rc_i}{\sum_{0}^{n-1} Rc_i} \times 100\%\right)\Big|_t, \quad (1)$$

Here, Ur_i is the utilization of i^{th} router, Rc_i is the register count of the $\mathbf{i^{th}}$ router, n is the total number of routers and **t** is the given simulation cycle time at which function call is being made.

Based on this criterion, it is observed (**Fig. 15- Global router utilization**) that for any given application workload, all the routers on the network could be classified into one of the following three categories: high, medium or low utilization. Majority of the routers fell into the medium utilization zone while very few were in either high or low utilization zones. It is realized that the low utilization zone is simply consuming energy while not doing much work for most of the time. Hence, as next step, an attempt is made to minimize unnecessary dissipation of power.

5.5. Dynamic Voltage Scaling (DVS)

The utilization metric tells that not all the routers are getting utilized at all times while overall, some routers are highly utilized while some are low utilized while majority fall under medium utilization. It is hence inferred that for any given application, it is not necessary to operate the router node at its full power. Depending on the need at given instant of time, the operating conditions of the router node can be varied as per the situation. This directly results in energy savings. The operating voltage is made variable and DVS implemented at router level. It is worth noting that DVS is implemented locally which implies that each of the router nodes have can have variable supply voltage instead of centrally varying the supply voltage for all nodes uniformly. Locality brings with it its own benefits and again reflects and hence contributed to the heterogeneity. Global configuration aligns all routers to follow identical thresholds and voltage scaling, while local configuration permits same on a per-core basis. Such flexibility allows creation of different sub-nets or clusters on the same network which will be significantly beneficial in case of large system size where activity of one cluster is likely to be less influenced by that of a far located one.

In order to vary the operating voltage, it is necessary to decide when to vary and by how much. The first part is taken care by making system calls to utilization function at a user defined epoch. The entire simulation period is divided equally into regular intervals called epoch. At the end of each epoch, the utilization function is called to determine the utilization of router. Using this value, the utilization of next epoch is estimated and supply scaled accordingly. It should be

noted that switching at every cycle will lead to higher transients hence it is best used at optimum period. A larger epoch will render such approach useless as it will be used once or twice during entire simulation while a very small epoch will give rise to transients. The second part uses user defined values for thresholds and voltages. A special case of DVS is to power gate the router node for supply falling to zero. Packet drop is drawback for such nodes. In case of minimal traffic to power gated router, packet drop is acceptable but this may not be case for a higher amount of traffic. In this case, router node by-pass channels need to go in place. Routing algorithm also needs to be modified at power-gated nodes to accept packet if the node is destination node. DVS implementation algorithm is given below.

Algorithm: Dynamic Voltage Scaling

At start of every $\mathbf{E_p}$ and given router node $\mathbf{R_i}$:
  If $\mathbf{Th_b} <= \mathbf{Ur_i} <= \mathbf{Th_a}$:
    $\mathbf{Vr_i} = \mathbf{V_a}$;
  Else If $\mathbf{Th_c} <= \mathbf{Ur_i} <= \mathbf{Th_b}$:
    $\mathbf{Vr_i} = \mathbf{V_b}$;
  Else If $\mathbf{Th_d} <= \mathbf{Ur_i} <= \mathbf{Th_c}$:
    $\mathbf{Vr_i} = \mathbf{V_c}$;
  Else:
    $\mathbf{Vr_i} = \mathbf{V_d}$;

Where, $\mathbf{R_i}$ is $\mathbf{i^{th}}$ router node, $\mathbf{E_p}$ is Epoch Period, $\mathbf{Ur_i}$ is utilization of $\mathbf{i^{th}}$ router, $\mathbf{Vr_i}$ is scaled voltage for $\mathbf{i^{th}}$ router and $\mathbf{Th_{a,b,c,d}}$ are utilization thresholds and $\mathbf{Vr_{a,b,c,d}}$ voltage levels set by user.

5.6. Floorplan-Aware Extension

In order to make this framework floorplan-aware, the network engine developed using above points can be integrated with a floorplan engine with ease, if need be. This not only makes the framework extensible to further developments but also aids in design taking into account the insights provided by the floorplan engine.

Impact of Floorplan on NoC is explained next. The floorplan of a logic block decides the physical placement of different chip components on the silicon and this significantly impacts chip area, power, thermal and reliability, etc. For network elements like router, links, etc. the floorplan impacts cost of communication in terms of latency and power consumption. Depending on the position of two processing elements and corresponding routers, the length of physical links and number of hops are determined. This directly impacts the

delay and energy required to complete a packet transfer between these two elements. This is particularly more important for core-to-memory access and core-to-core communication in heterogeneous systems where physical size of cores is also different. In a homogeneous multi-core system, irrespective of the placement of cores, core-to-core communication can be optimized by efficient application mapping techniques. This is worsened in case of heterogeneous systems due to difference in link bandwidth and length depending on core to which it is connected together with the physical location of the elements. It is desirable from communication performance perspective that two frequently communicating elements be placed near to each other. But, the combined high activity of these two elements can lead to thermal hotspots because of heat spread between each other. Extension to a pre-RTL floorplanning engine that determines the layout based on individual component areas, power consumption and thermal stability is proposed. Cost of communication between two node pairs is used as input for this extension to evaluate performance of NoC for different floorplan layouts of the same chip. This engine is based on simulated-annealing [36] and illustrated in Fig. 5. Thermal profiles, cost-of-communication between two cores and hotspot thresholds are taken as inputs. These constraints and attributes guide the engine to generate acceptable floorplans.

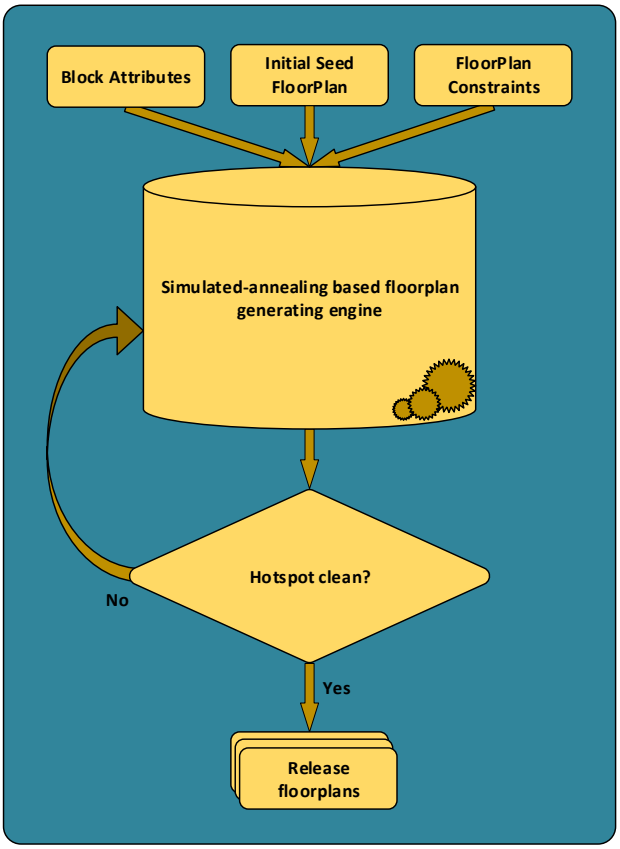


Fig. 5- Floorplan-aware extension flowchart

5.7. How to use this framework?

The overall flow for the proposed evaluation framework including full-system simulation is illustrated in Fig. 6.

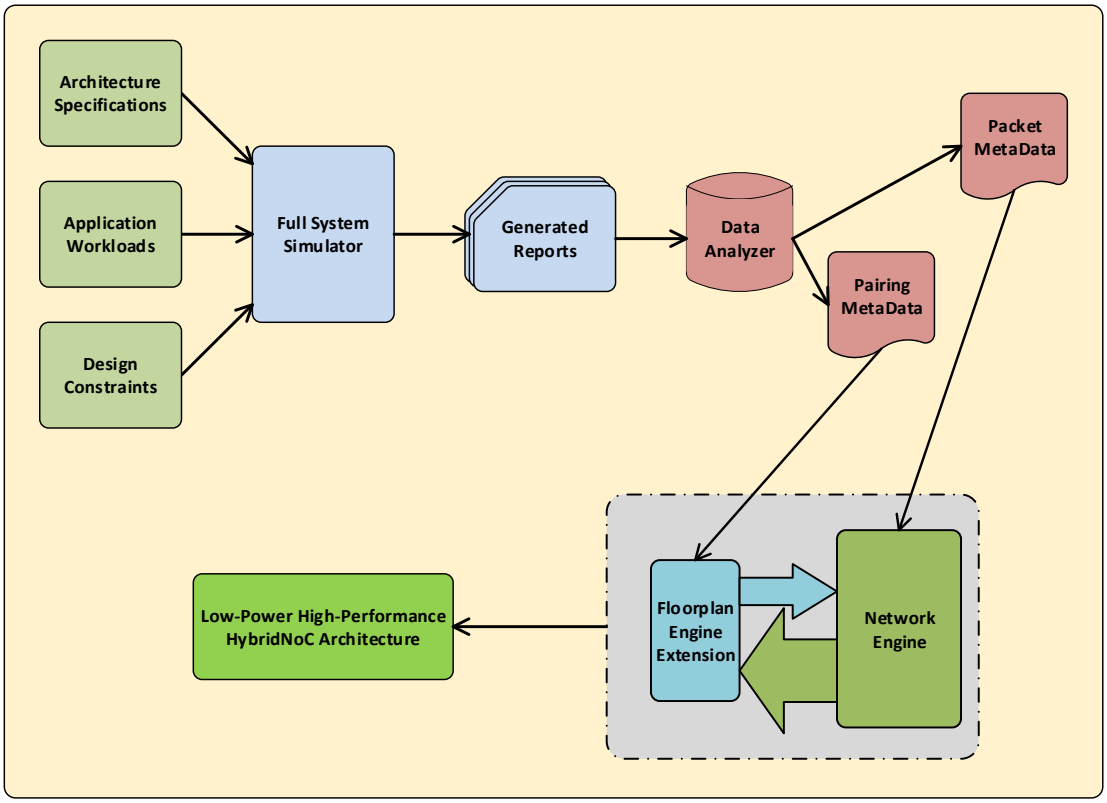


Fig. 6- Overall evaluation framework flow

The full-system requires architecture specifications and design constraints. Simulations generate numerous reports for given application workload. Data Analyzer, a Python based tool is specifically developed for this purpose to analyze and parse the huge amount of data so generated. Output of this tool generates two specific files. The first file, Packet Metadata, contains cycle accurate information regarding packet movement inside the full system simulator. This forms the basis for network simulations which is carried out in the next step. The second file, Pairing Metadata, contains information regarding cost attached with communication between any two nodes on the system. The next step is to carry out network simulations to iteratively arrive at the optimum topology.

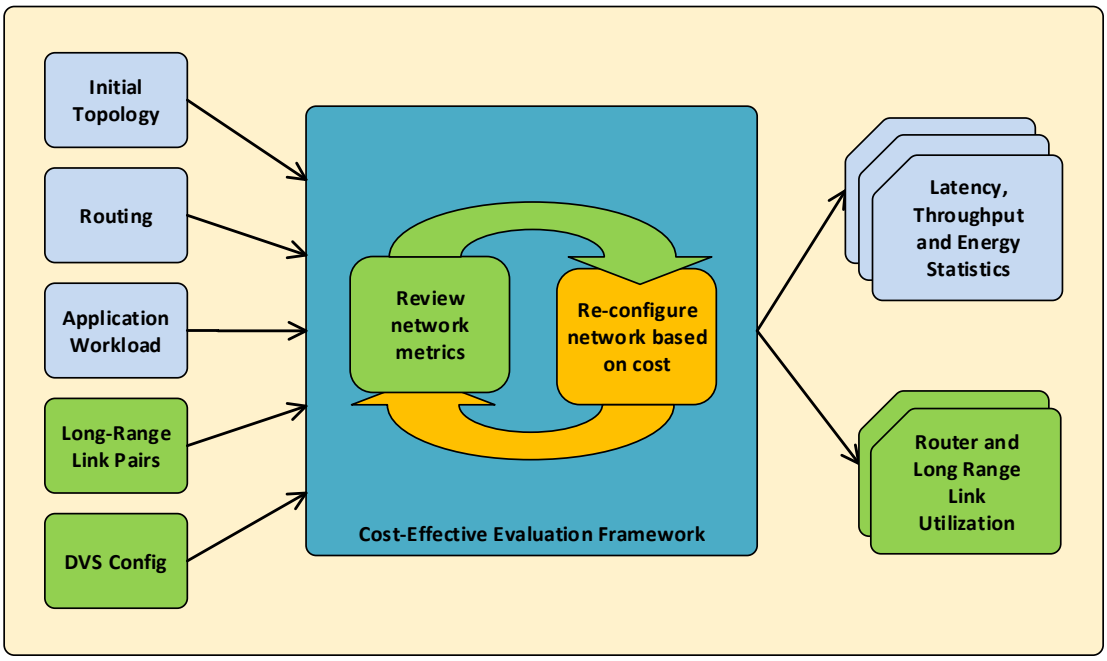


Fig. 7- Network evaluation flow

Fig. 7 illustrates the network evaluation flow. This requires selecting an open-source network simulator as base and implementing modules depicted in green on top of it to enhance it into a cost-effective evaluation framework. The base already contains modules depicted in light blue. For illustration purpose, let the initial topology be regular 2D-mesh which when augmented with long-range links results in a hybrid topology. The optimum number and placement of long-range link pairs are obtained from exhaustive-search based, MATLAB algorithms which takes system configuration and traffic distribution into consideration. With relevant routing information and traffic pattern as input, this first-cut topology is evaluated for network performance. Based on performance statistics and utilization, the number and placement of long-range link pairs should be re-configured manually by re-writing the long-range link pair text file. This process is iterative and is recommended to gradually obtain high-performance hybrid topology. The MATLAB based algorithm and manual re-configurations can also be automated by implementing them within the network flow itself. In this case, a corresponding threshold should be introduced and fed as input to aid decision making, avoid unnecessary iterations and hence avoid huge runtimes. DVS is enabled to design and evaluate energy-efficient architectures. Performance of hybrid topology is expected to be better than regular mesh topology but the addition of extra hardware may lead to thermal stability issues. To probe one of many such drawbacks, the block layout for the topology generated from network flow, thermal profile and block attributes of nodes and Pairing MetaData is fed to the floorplan engine to check for any violations. Because the network topology is not floorplan aware, it is very likely that this layout will highlight severe hotspots. This engine then incrementally generates different sets of

floorplans for all IPs on the system. The updated block positions are then mapped to corresponding nodes on the NoC. With changes in node mapping, the existing topology may no longer hold good and hence the topology evaluation step is again exercised. After few iterations, the hybrid NoC topology so generated is not only high-performance and optimized for energy-efficiency (only if DVS is enabled)but is also floorplan aware. It is again emphasized that each of the modules can be enabled as per need and corresponding input variables are all user configurable. This approach makes the framework very robust and extensible to further development.

6. Implementation and Results

The proposed framework outlined in chapter five is implemented on top of Noxim, a cycle accurate SystemC based standalone network simulator available as open source. Noxim supports only regular 2D-mesh (both, symmetric and non-symmetric), table and algorithmic based routing, synthetic and table based traffic patterns, range of network packet-injection-rate (PIR), probability-of-retransmission (POR), buffer-depth and packet size configurability, wormhole flow control, etc. Hence, based on existing features, Noxim emerges as best choice for implementing the proposed framework. However, the framework can be implemented on any other network simulator with equal ease.

Note: This framework is a result of the requirements of research work being carried out in the Advanced Multicore Systems Lab, IIIT-D. Inputs from framework modeling and development has significantly influenced the design of energy-efficient routers and hybrid NoC architectures hence proposed. Detailed real time insights obtained during simulation by enabling the utilization metric is the basis for these proposals while the low-power techniques are implemented and also evaluated using this same framework. However, the inherent complexity attached with these proposals involve other aspects, which includes hybrid NoC system design, router internals digital circuit design, register transfer level (RTL) router implementation for area and power analysis, wireless channel modelling, high frequency transmitter-receiver transistor level design for wireless communication, network framework development, simulation tools integration, design and performance validation of proposed topologies, etc. Proceedings from this collaborative research work are presented as case studies in this chapter to emphasize the significance of the network evaluation framework developed in this thesis for current trends in NoC research.

Common setup configuration for the implemented framework is described next. The detailed simulation setup is given in each of the case studies respectively.

Traffic trace: Network simulators do not support direct execution of application benchmarks. Full-system simulators are required for this purpose. GEM5 full-system simulator is used to configure system for 16-cores and 64-cores, as required. It generates large number of reports. Traffic information in these reports cannot be directly used as input to base Noxim. Hence, a Python based tool 'Data Analyzer' is developed to generate 'Packet Metadata' Table 2 from Table 1 which is compatible with existing table based traffic input. Full-system trace that does not include cache-to-cache information is eliminated. Destination ID (DST_ID) with value -1 indicates broadcast message. The network PIR and POR is set to 1 due to the availability of unique timestamps for each trace entry. And the need to release this unique packet at unique time in the network also. However, in case of non-availability of timestamps or availability of node pair communication in form of a probability, then a lower PIR and POR can be set and the network behavior studied.

Table 1- Sample full-system traffic trace

Timestamp	SRC	<SRC_ID>	DST	<DST_ID>	Msg_Type	Address
36000	src	56	dst	-1	RequestMsg	0x1fa48a00
36500	src	56	dst	-1	RequestMsg	0x1fa48a00
37000	src	12	dst	56	ResponseMsg	0x1fa48a00
37500	src	28	dst	56	ResponseMsg	0x1fa48a00

Table 2- Packet Metadata generated by Data Analyzer

SRC	DST	PIR	POR	Timestamp
12	56	1	1	37000
28	56	1	1	37500

Hybrid Topology: It contains information about long-range link pairs. The framework reads this file and automatically creates and links the router nodes with long-range links. To re-configure the network, this file needs to be updated. Format of this file is depicted in Table 3.

Table 3- Long-range links file format

LinkID	SRC	DST
0	5	13
1	9	11

Hybrid Routing: This file contains routing information for both regular and hybrid routers. For case studies under consideration, XY routing is enabled for

regular routers. But Incoming packet from hybrid port may result in cyclic dependency thereby posing potential deadlock problem. Hence, North-Last routing is used for hybrid routers. This is adopted from small-world on-chip networks [5] wherein a turn-model based theorem and proof is also presented for deadlock and livelock free routing. Additional routing information is mentioned in respective case studies.

DVS configuration: This is used to enable dynamic voltage scaling. Nominal values of thresholds and voltages are mentioned in respective case studies. These values are similar to the ones used in [16]. Epoch period of 1000 cycles is used for all setups and is also set using this option.
For special cases like power-gating only, the voltage levels are limited to only two levels: 0 and 1 (off and on respectively). Such configuration also enables evaluation of networks with different clusters operating with minimum dependency on each other. An example of such a system is the modern quad core cluster with each core having its own memory sub-system but configured on the same network. This enables multi-tasking by mapping different applications on the available cores. Subsequent cluster-based energy analysis can now be carried out and the findings contrasted with traditional full chip methods. Cluster-based configuration is only presented as potential problem to be analyzed using this framework and is not part of the case studies under consideration.

Technology Specifications: This requires energy dissipation information for hybrid links implemented. For case studies, the system is operating at 2.5GHz [27] and all long-range links are single-hop. RTL level router design is implemented in 28nm CMOS technology using Synopsys tools. Power and area analysis is carried out using Cadence tools.

6.1. Case Study I: Hybrid NoC Topology[1]

The purpose of this case study is to validate credibility of the implemented framework. A regular 2D-mesh is augmented with long-range links to get a hybrid NoC. These links are obtained using exhaustive search algorithms considering the various synthetic traffic the network is expected to handle. Performance is compared with regular 2D-mesh topology.

Configuration: Simulation setup is given in table below.

Table 4- Case Study I: Simulation Setup

Topology	8x8 Mesh | 8x8 Hybrid NoC
Routing	XY for regular | North-Last for hybrid routers
Arbiter	Round-robin
Flit size	32-bits
Packet size	64-flits
Long-range link	Single-hop | Wireless | Directional
Clock frequency	2.5GHz
Workload	Synthetic

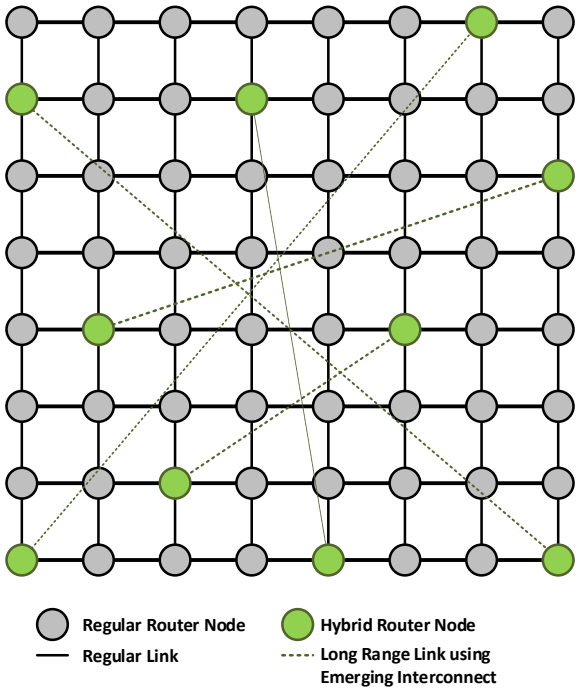
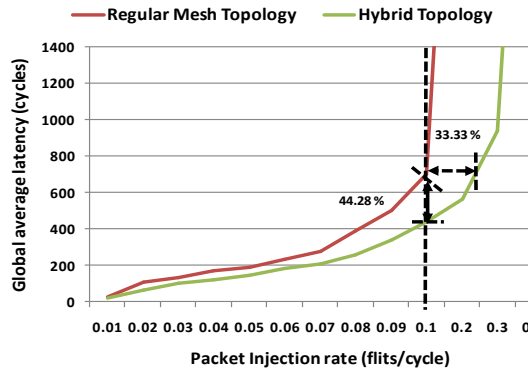


Fig. 8- Hybrid NoC Topology with long-range links

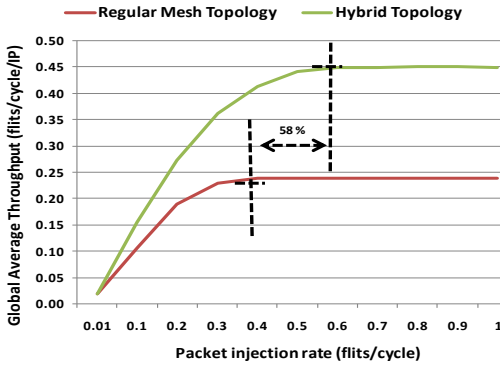
[1] Adapted from the paper accepted at VDAT-2016, Guwahati, India [1]

Performance Analysis: For transpose traffic, the network packet latency at 0.095flits/cycle drops from 720 to 430 cycles, providing about 40.28% reduction and the phase transition region shifts from a throughput of 0.34flits/cycle to 0.54flits/cycle, giving 58% saturation throughput improvement. Long-range wireless links shifts critical loads from 0.15 to 0.2-flits/cycle, giving about 33.33% improvement. Significant improvement in network performance is observed using the hybrid architecture with long-range links over regular mesh architecture (Fig. 9).

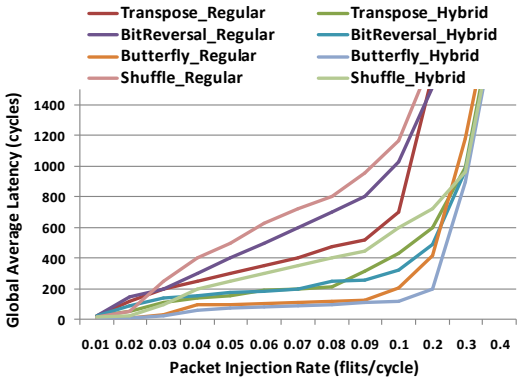
It needs to be noted that the initial number of links generated by the exhaustive search algorithm was six. Analysis of the Packet MetaData file lead to selective changes in the positions of the long-range links. The topology Fig. 8 shown is finalized only after few iterations.



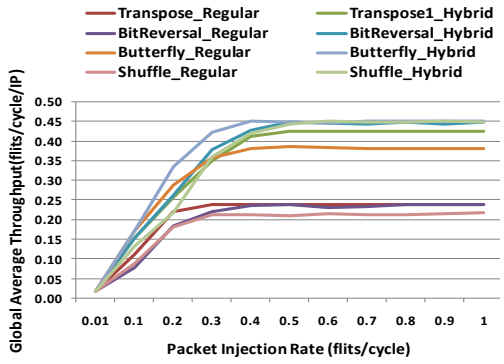
a) Average latency for transpose



b) Average throughput for transpose



c) Average latency for synthetic traffic



d) Average throughput for synthetic traffic

Fig. 9- Network performance

Inference: The long-range links are created by writing link node pairs to text file which is input to the framework. Re-configurations requires to simply update the input long-range links text file. This demonstrates the immense utility, user friendliness and efficiency of the evaluation framework as it enables quick configuration and evaluation of any topology.

The performance improvements observed in hybrid topology over regular mesh are coherent with those generally observed results for similar topologies. This validates the credibility of the implemented framework.

6.2. Case Study II: Network Utilization of Small-World On-Chip Topology with Directional Links[2]

The purpose of this case study is to demonstrate the usefulness of the utilization metric introduced in this framework. Popular small-world on-chip network, which is based on the philosophy of six-degrees of freedom, is configured. The topology essentially decreases the NoC diameter significantly. Utilization analysis is carried out for traffic patterns different from the ones the topology is originally proposed for. This is to illustrate that the network performance is dependent on traffic conditions which is a function of PIR, packet size, traffic pattern, congestion, etc. It may not be necessary that the long-range links may always be beneficial at all times. Therefore a mechanism is required to monitor the actual activity of the added links and also of the routers.

Configuration: Simulation setup is given in Table 5.

Table 5- Case Study II: Simulation Setup

Topology	4x4 Mesh | 4x4 Hybrid NoC
Routing	XY for regular | North-Last for hybrid routers
Arbiter	Round-robin
Flit size	32-bits
Packet size	64-flits
Long-range link	Single-hop | Wireless | Directional
Clock frequency	2.5GHz
Workload	Synthetic
Utilization	Enabled

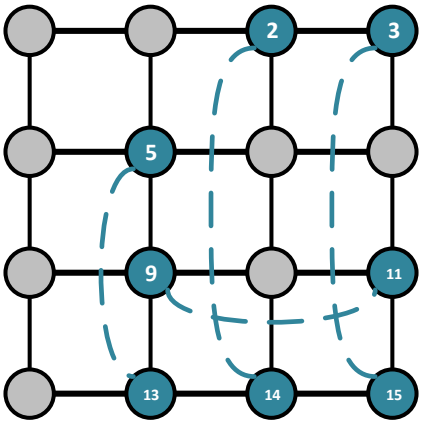


Fig. 10- Small world on-chip NoC topology with directional links

[2] Adapted from the paper accepted at VDAT-2016, Guwahati, India [1]

Performance Analysis: The hybrid topology as compared to regular mesh topology results in over 50% improvement in packet latency with over 37% improvement in packet throughput. The network saturation phase transition point shifts from 0.175 packet-injection-rate to 0.275 packet-injection-rate. Peak utilization of directional wireless links inserted is over 30% when compared to total utilization of both regular and irregular links. Achieved results demonstrate significant improvements in network statistics over regular mesh topology and are coherent with those generally observed in small-world on-chip networks.

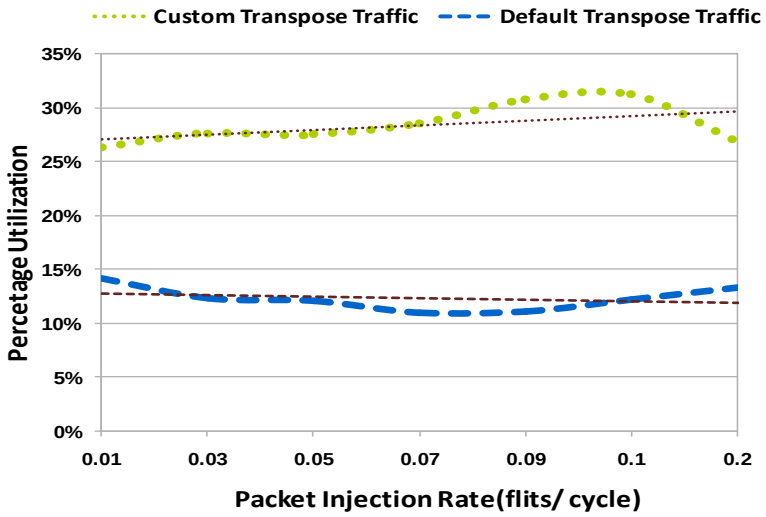


Fig. 11- Wireless link utilization for different transpose traffic

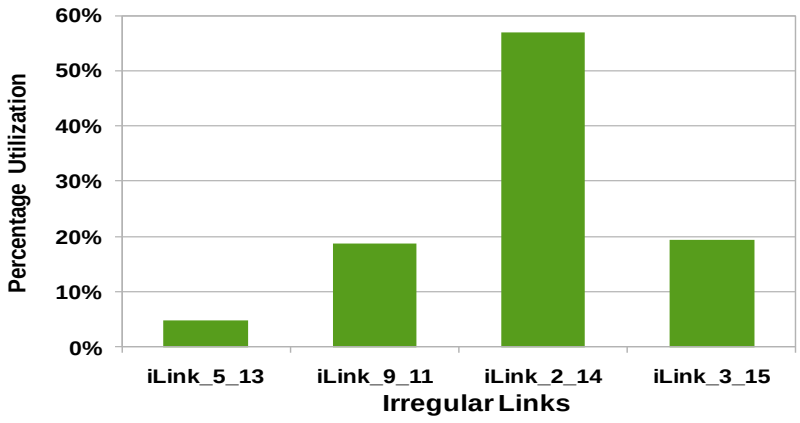


Fig. 12- Individual long-range link usage @ 30% overall utilization for custom transpose traffic

Table 6- Variable number of WI for shuffle traffic

	Mesh	iLinks = 1	iLinks = 2	iLinks = 3	iLinks = 4
iLink_5_13	-	100%	26%	33%	23%
iLink_2_14	-	-	73%	44%	24%
iLink_3_15	-	-	-	22%	15%
iLink_9_11	-	-	-	-	37%
Latency	1.00	0.37	0.24	0.15	0.20
Throughput	1.00	1.20	1.19	1.15	1.18
Energy	1.00	1.10	1.05	1.07	1.10

Design space exploration with variable number of wireless interface (WI) pairs (Table 6) indicates that for given configuration, it is not necessary to use all four irregular links which is contrary to expectations. Mainly, this is because different network conditions get formed due to different traffic patterns and injected load. However, this can be interpreted in two different ways. First, the highly parallel nature of network simulations may not be accurately captured by exhaustive search algorithms used to generate long-range link pairs, even for traffic it is intended for. This is reflected in real time by the utilization metric. Some links may be heavily used while others rarely used. Second, using all four links should result in best values of network latency, throughput and packet energy but because the topology is evaluated for traffic pattern different than what it was intended for, significant deviations are observed. In this case, three links give good performance hence the fourth link can be omitted thereby reducing cost. The utilization of wireless link W_5_13 is different in the Fig. 12 and Table 6 because they are quoted for two different network conditions, transpose and shuffle respectively. Irrespective of why the results are not as per initial estimates, in any of the cases, link utilization is that mechanism which can be used to optimize the topology for application-specific requirements.

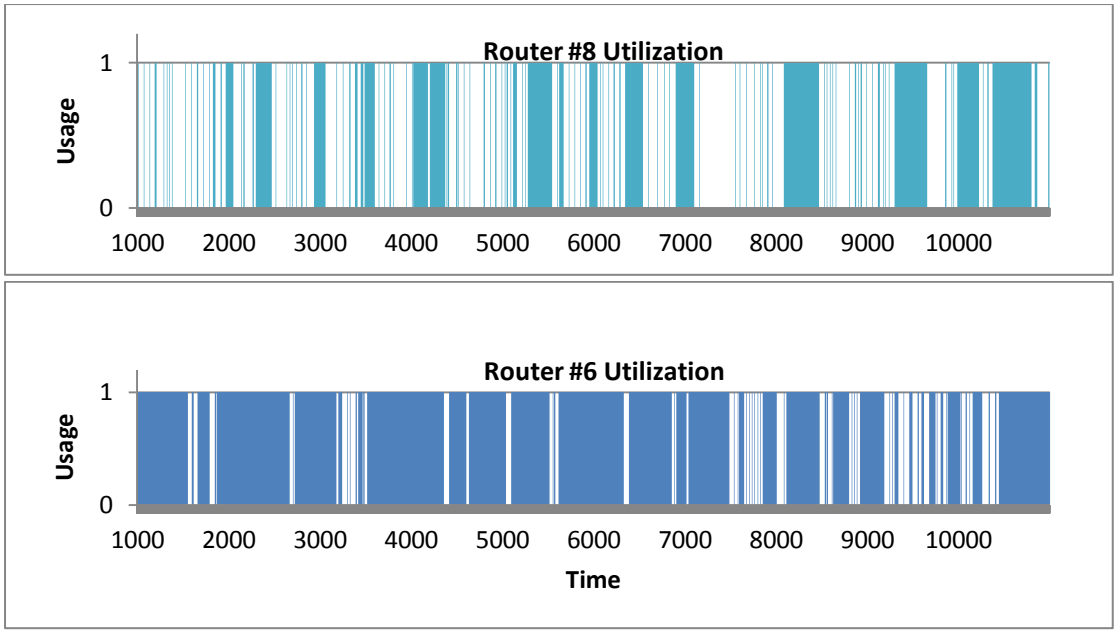


Fig. 13- Real time utilization of routers

On any NoC, not all routers will be active all the time. This is observed from the router utilization plots. Fig. 13 indicates that router #6 is heavily utilized as compared to router #8. White spaces indicate the time periods during with both the routers are idle thereby consuming unnecessary energy. This information can be used to apply power saving techniques and is undertaken in detail in subsequent case studies.

Inference: The usefulness of the utilization framework is apparent from the various plots above. The insight provided for both routers and long-range links demonstrates the increased confidence a designer will have while proposing a hybrid architecture because it is real time information which enables to further optimize the topology which can be done with ease using this framework for application-specific requirements. The utilization metric quantifies the usefulness of the hardware implemented in real time, highlights the cost-effective nature of the framework and can now be exploited to implement energy-saving techniques.

6.3. Case Study III: Energy-Efficient Wireless NoC Architecture I[3]

Purpose of this case study is to highlight the role of the framework in design and evaluation of energy-efficient wireless NoC architectures, the basis of which is the utilization metric. Router static power dissipation is reduced by application of power-gating. To minimize packet drop due to power-gated router nodes in forward path, by-pass channels are inserted and performance evaluated. These techniques are implemented on Wireless NoC (WNoC) with omni-directional links instead of directional links used in previous case studies. Omni-directional links will be discussed in detail in next case study.

Configuration: Simulation setup is given in Table 7 and in Table 8.

Table 7- Case Study III: Simulation Setup

Topology	4x4 Mesh | 4x4 Hybrid NoC
Routing	XY for regular | North-Last for hybrid routers | Routing modified to use bypass channels and omni-directional links
Arbiter	Round-robin
Flit size	32-bits
Packet size	64-flits
Long-range link	Single-hop | Wireless | Omni-directional
Clock frequency	2.5GHz
Workload	Synthetic and real benchmarks
Utilization	Enabled
DVS	Enabled with bypass channels

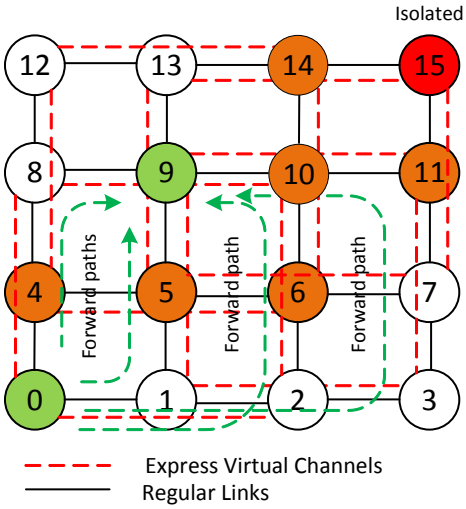


Fig. 14- Power-gated Topology

[3] Adapted from the papers accepted at IGSC-2016, Las Vegas, US [2] and at ISQED-2016, Santa Clara, US [3]

Based on observations in Case Study II, for given application workload, network routers are broadly classified into three zones: High Utilization Zone (HUZ), Medium Utilization Zone (MUZ) and Rare Utilization Zone (RUZ) Fig. 15. This classification is global, i.e. it is carried out only once before the start of the network simulation. In this architecture, power-gating is applied both globally as well as locally. Globally, all routers pre-determined to be falling in RUZ are permanently power-gated. Locally, the routers are switched on and off as per their real time utilization (Table 8). For this, at end of every user-defined epoch period, a function call is made to calculate the utilization during that epoch and is used to predict the router utilization in next epoch. Switching thresholds used are based on analysis of average router utilization values across different application benchmarks. This proposed Fine-Grained Router Architecture (FGRA) is power-aware and is intended for application-specific requirements. For detailed circuit-level hardware implementation, the interested reader is referred to the published papers.

Table 8- Case Study III: DVS configuration for power-gating

Thresholds*		**Voltages***	
Th_a	>5%	**V_a**	1.0
Th_b	<=5%	**V_b**	0.0

*All values are normalized

Main drawback of power-gating is packet drop. This degrades network performance. Fig. 14 depicts typical corner case in which the globally power-gated routers (orange) block forward path of packet (where routers white, red and green are awake). To facilitate path for packet to move from router-0 to router-9, by-pass channels or express virtual channels [37] are implemented. The routing algorithm is modified to use these bypass-channels and also ensures that in case the power-gated router is itself the packet destination, then the incoming packet is not dropped.

Performance Analysis: The energy savings obtained by FGRA is evaluated by running PARSEC and SPLASH-2 benchmarks on 16 core system and results are presented in Fig. 16. The energy savings are over a regular WNoC architecture without power gating. On average, FGRA achieves 37.20% savings in static energy across all benchmark as compared with regular WNoC. The energy saving using FGRA reaches above 40% for data intensive applications like Canneal. The comparison is made between the WNoC with and without the proposed router. To evaluate performance under both computation and communication intensive

workloads, SPLASH-2 and PARSEC benchmarks are considered. Fig. 17 shows peak bandwidth (peak bandwidth is the maximum achievable data rate for the WNoC) at network saturation. FGRA improves or performs equally well in more than half the tested benchmarks and traffic patterns. Even the performance degradation observed with Canneal, Swaptions and FFT traffic is negligible compared to regular architecture. This utilization based approach can save static power consumption upto 88.76% in base router (BR) and 62.50% in hybrid router.

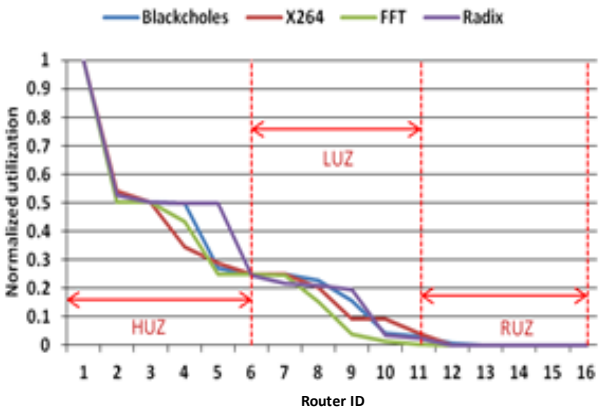


Fig. 15- Global router utilization

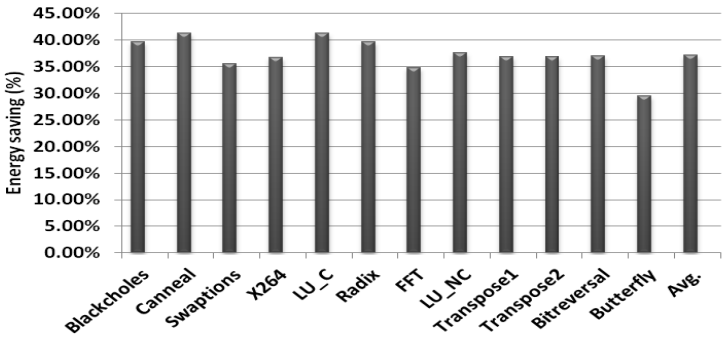


Fig. 16- Energy savings

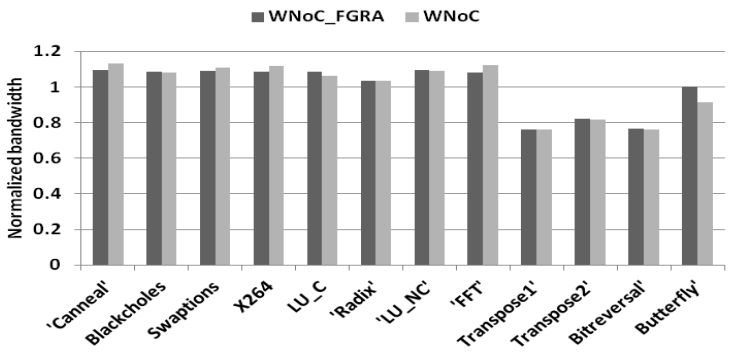


Fig. 17- Normalized bandwidth

Table 9-Comparison with existing energy-efficient architectures [2][3]

Approaches	**Power/ Energy saving (%)**	**Penalty**
Power Punch	Router static energy: 83%	0.4% execution Time
FlexiBuffer	Overall router power: 39%	3% degradation in throughput
NoRD	Router static energy:29.9%	3% area overhead
FGRA	Router static power: 88.76% (BR); 62.5% (WI)	2.42% area overhead

Inference: Basis for FGRA proposal is utilization. Different levels of power-gating is implemented using DVS feature in the framework. User-configurable thresholds, voltages and epoch period illustrates the ability of the framework to design and evaluate energy-efficient hybrid NoC architecture. The scope for design space exploration is widened by not limiting to only pre-defined values. Agility and robustness of the framework is demonstrated by the ability to insert bypass channels with minimum difficulty.

6.4. Case Study IV: Energy-Efficient Wireless NoC Architecture II[4]

In previous case study, power-gating was applied by fully turning off the routers. Some routers may have activity higher than power-gating threshold but may still not require full operating conditions due to intermediate activity. In this case study, both static as well as dynamic router power is reduced by using four different low-power modes (Table 11).

Configuration: Simulation setup is given in Table 10 and Table 11.

Table 10- Case Study IV: Simulation Setup

Topology	4x4 Mesh | 4x4 Hybrid NoC
Routing	XY for regular | North-Last for hybrid routers | Routing modified to use omni-directional links
Arbiter	Round-robin
Flit size	32-bits
Packet size	64-flits
Long-range link	Single-hop | Wireless | Omni-directional
Clock frequency	2.5GHz
Workload	Synthetic and real benchmarks
Utilization	Enabled
DVS	Enabled

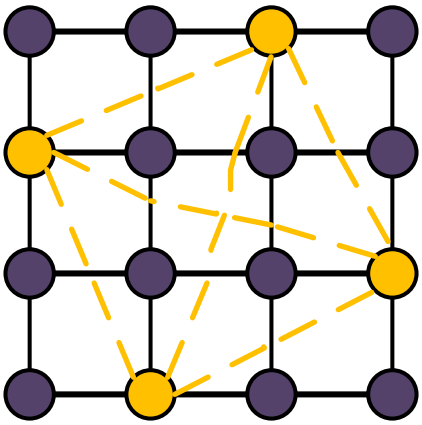


Fig. 18- Wireless NoC with omni-directional links

Considering the dynamic nature of network activity, the directional links previously used are now replaced by omni-directional links. This enables the hybrid router node to pair and communicate with any other hybrid router node

[4] Adapted from the paper accepted at DATE-2016, Dresden, Germany [4]

as per network conditions. Such complex setup is expected to improve system performance. In Fig. 18 the regular (purple) nodes have fixed linkages whereas the hybrid (yellow) are omni-directional.

Table 11- Case Study IV: DVS configuration for AMS

Thresholds*		**Voltages***	
$\mathbf{Th_a}$	100%	$\mathbf{V_a}$	1.2
$\mathbf{Th_b}$	75%	$\mathbf{V_b}$	1.0
$\mathbf{Th_c}$	25%	$\mathbf{V_c}$	0.8
$\mathbf{Th_d}$	5%	$\mathbf{V_d}$	0.0

*All values are normalized

Performance Analysis: The proposed Adaptive Multi-Voltage Scaling (AMS) is also utilization-based approach (Table 11). The procedure to scale router to ascertained voltage level also uses function calls to utilization function. This approach saves static power consumption up to 62.50% in hybrid router and the overall packet energy consumption by 35% (Fig. 19). Omni-directional wireless links enable a higher global average throughput compared with baseline architectures. The variation of network throughput as a function of packet injection rate is plotted in Fig. 20. Random (or uniform random) traffic shows abnormally high packet energy savings because it treats all nodes with equal probability. This is not true in case of other synthetic and real workloads.

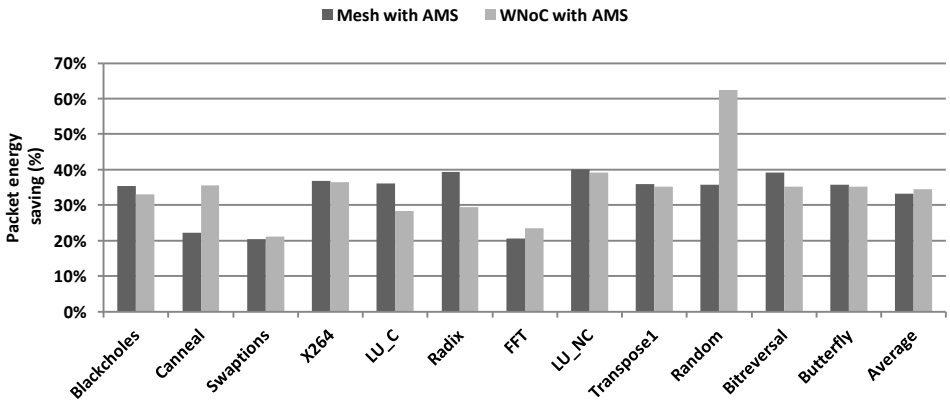


Fig. 19- Energy savings

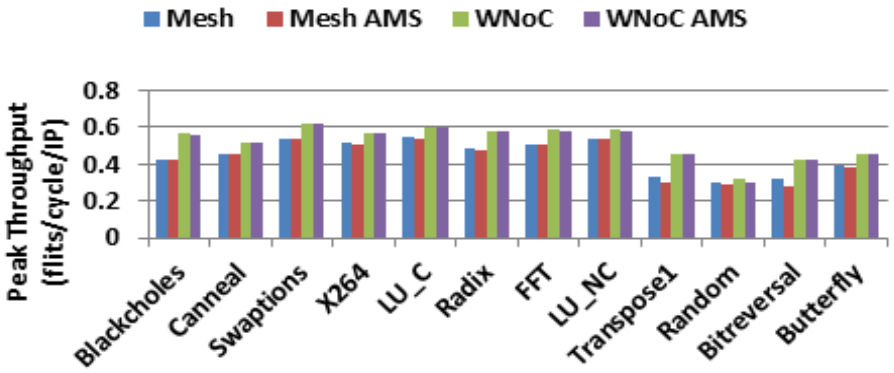


Fig. 20- Peak throughput

Table 12-Comparison with existing energy-efficient NoC architectures [4]

Approaches	**Power/ Energy saving (%)**	**Penalty**
Pruning DVFS in WNoC	Energy-delay: 24.80%	10% Area overhead with WI
DVFS-enabled sustainable WNoC	Overall energy: 60%	---
DVS policy based on link utilization	Energy-delay: 36%	Negligible performance degradation
Power Punch	Router static energy: 83%	0.4% execution Time
Panthre	Overall network power: 14.5%	1.8% degradation in performance
FlexiBuffer	Overall router power: 39%%	3% degradation in throughput
NoRD	Static energy per router:29.9%	3% area overhead
Virtual channel power-gating	Overall static power: 40%	0.3% throughput degradation
AMS-based WNoC	Overall packet energy: 20% - 62.43%; Static power per WI : 62.50%	Less than 1% area overhead

Inference: Although the framework was not intended for omni-directional links, it was still possible to model and evaluate them due to the robustness and flexibility of the framework to evaluate more complex hybrid architecture with minimal difficulty. The routing algorithm for omni-directional links was also implemented with ease. This is possible only due to the user-defined nature of inputs for topology, routing, etc. enabled in the framework. The proposed architecture implemented is also energy-efficient.

7. Conclusion and Future Work

In this thesis, an efficient, robust and cost-effective evaluation framework is proposed and implemented to design and evaluate hybrid NoC architecture. The insight provided by network utilization is exploited to apply DVS with power-gating to realize an energy-efficient architecture. Case studies undertaken demonstrate the ease of re-configurability thereby highlighting the usefulness and the agility of the framework. Performance evaluation of various hybrid architectures indicate significant improvements in network performance over regular mesh topology with and without DVS thereby vindicating the benefits of using hybrid architecture for multicore systems.

Key contributions of this work include:

- Cost-effective utilization based efficient and user re-configurable evaluation framework is developed
- Quick and user-friendly evaluation framework, which can be used with ease
- Extensible framework enabling further development with minimum difficulty

Future work includes thermal stability study of heterogeneous multicore system topologies using the proposed floorplan engine extension and the subsequent release of the proposed framework in this thesis as open source.

List of Publications

[1] **Raghav Kishore**, Hemanta Kumar Mondal and Sujay Deb, "Energy-efficient Reconfigurable Framework for Evaluating Hybrid NoCs," **VDAT** 2016, India [Accepted]

[2] Hemanta Kumar Mondal, Sri Harsha Gade, **Raghav Kishore** and Sujay Deb, "Adaptive Multi-Voltage Scaling in Wireless NoC for High Performance Low Power Applications", **DATE** 2016, Germany

[3] Hemanta Kumar Mondal, Sri Harsha Gade, **Raghav Kishore**, Shashwat Kaushik and Sujay Deb, "Power Efficient Router Architecture for Wireless Network-on-Chip," **ISQED** 2016, Santa Clara

[4] Hemanta Kumar Mondal, Sri Harsha Gade, **Raghav Kishore** and Sujay Deb, "Power- and Performance-Aware Fine-Grained Reconfigurable Router Architecture for NoC" **IGSC** 2015, Las Vegas

References

[1] http://spectrum.ieee.org/tech-talk/computing/hardware/how-livermore-scientists-will-put-ibms-brain-inspired-chips-to-the-test

[2] http://developer.amd.com/resources/heterogeneous-computing/

[3] http://www.hsafoundation.com/

[4] Marculesco, et al., “Outstanding Research Problems in NoC Design: System, Microarchitecure and Circuit Perspecetive”, IEEE Trans. on CAD-ICS, vol. 28, No.1, January 2009, pg. 3-21

[5] U. Ogras and R. Marculesco, “It’s a Small World After All: NoC Performance Optimization Via Long-Range Link Insertion”, IEEE Trans. VLSI, Vol. 14, No.7, July 2006

[6] Das, et al. “Design and Evaluation of a Hierarchical On-Chip Interconnect for Next Generation CMPs,” IEEE 15th International Symposium on HPCA, 2009

[7] L. Carloni, P. Pande and Y. Xie, “Networks-on-Chip in Emerging Interconnect Paradigms: Advantages and Challenges”, NOCS 2009

[8] Binkert et al., "The GEM5 Simulator," ACM SIGARCH Comp. Arch. News, vol. 39, no. 2, pp. 1-7, August 2011

[9] Rafael Ubal, Byunghyun Jang, Perhaad Mistry, Dana Schaa, and David Kaeli, "Multi2Sim: A Simulation Framework for CPU-GPU Computing", 21st International Conference on Parallel Architectures and Compilation Techniques, Minneapolis, MN, September 2012

[10] https://github.com/davidepatti/noxim

[11] Nan Jiang, et al., “A Detailed and Flexible Cycle-Accurate Network-on-Chip Simulator” IEEE International Symposium on Performance Analysis of Systems and Software, 2013

[12] Tran, Anh T., and Bevan M. Baas. “NoCTweak: A highly parameterizable simulator for early exploration of performance and energy of networks on-chip”, Tech. rep., VLSI Computation Lab, 2012

[13] Hossain, Hemayet, et al. "Gpnocsim-A general purpose simulator for network-on-chip." International Conference on Information and Communication Technology(ICICT'07)., pp. 254-257. 2007

[14] Jain, Lavina, et al. "NIRGAM: a simulator for NoC interconnect routing and application modeling." In Design, Automation and Test (DATE), 2007.

[15] Prabhu, Subodh, ”Ocin_tsim-A DVFS Aware Simulator for NoC Design Space Exploration and Optimization”. Master's thesis, Texas A&M University, 2010

[16] D. Zoni, F. Terraneo and W. Fornaciari, “A DVFS Cycle Accurate Simultion Framework with Asynchrounous NoC Design for Power-Performance Optimizations,” Journal of Signal Processing Systems, 2015

[17] D. Zoni, W. Fornaciari, “Modeling DVFS and Power-Gating Actuators for Cycle-Accurate NoC-Based Simulators”, ACM Journal on Emerging Technologies in Computing Systems (JETC), September 2015

[18] L. Chen and T. Pinkston, “NoRD: Node-Router Decoupling for Effective Power-Gatingof On-chip ROuters” 45th Annual International Symposium on Microarchitecture, 2012

[19] R. Parikh, R. Das and V. Bertacco, “Power-Aware NoCs through Routing and Topology Reconfiguration,” Design Automation Conference (DAC), 2014

[20] Chen, et al, “DSENT – A Tool ConnecCng Emerging Photonics with Electronics for Opto-Electronic Networks-on-Chip Modeling”, NOCS 2012

[21] J. Miller, et al., “Graphite: Distributed Parallel Simulator for Multicores”, The 16th IEEE Symp. On HPCA 2010

[22] http://lava.cs.virginia.edu/HotSpot/

[23] G. Faust, et al, “ArchFP: Rapid Prototyping of pre-RTL Floorplans”, VLSI-SoC 2012

[24] Murali, et al., “Designing Application Specific Networks on Chips with Floorplan information”, ICCAD 2006

[25] Z. Wang, et al, “Floorplan Optimization of Fat-Tree-Based Networks-on-Chip for Chip Multiprocessors”, IEEE Transactions on Computers, Vol.63, No.6, June 2014

[26] A. Shacham, et al., “Photonic Network-on-Chip for Future Generations of Chip Multi-Processors,” IEEE Transactions on Computers, Vol. 57 No. 9 pp. 1246-1260, 2008

[27] S. Deb, et al., “Wireless NoC as Interconnection Backbone for Multicore Chips: Promises and Challenges,” IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS), Vol. 2 No.2 pp. 228-239 2012

[28] Abad, Pablo, et al. "Topaz: An open-source interconnection network simulator for chip multiprocessors and supercomputers." In Networks on Chip (NoCS), pp. 99-106, 2012

[29] Agarwal, Niket, et al. "GARNET: A detailed on-chip network model inside a full-system simulator." IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS) pp. 33-42, April, 2009

[30] Trevor E. Carlson; Wim Heirman; Lieven Eeckhout. Sniper: Exploring the Level of Abstraction for Scalable and Accurate Parallel Multi-Core Simulation. International Conference for High Performance Computing, Networking, Storage and Analysis (SC)

[31] Hestness, Joel, et al., "Netrace: dependency-driven trace-based network-on-chip simulation." ACM International Workshop on Network on Chip Architectures., pp. 31-36, 2010
[32] http://www.amd.com/en-us/innovations/software-technologies/apu
[33] PK Gupta, "Xeon+FPGA Platform for the Data Center," ISCA/ CARL 2015
[34] Altera Whitepaper, "Enabling High-Performance DSP Applications with Stratix V Variable-Precision DSP Blocks," May 2011
[35] http://www.eembc.org/
[36] http://www2.stat.duke.edu/~scs/Courses/Stat376/Papers/TemperAnneal/KirkpatrickAnnealScience1983.pdf
[37] Amit Kumar, et al., "Express virtual channels: towards the ideal interconnection fabric" SIGARCH Comput. Archit. News, pg.150-161, 2007