A Transmission Line Enabled Deadlock Free Toroidal Network-on-Chip using Asynchronous Handshake Protocols

Mackenzie J. Wibbels[1], Shomit Das[2], Dheeraj Singh Takur[2], Venkata Nori[1], and Kenneth S. Stevens[1]

[1]University of Utah
[2]Advanced Micro Devices, Inc.

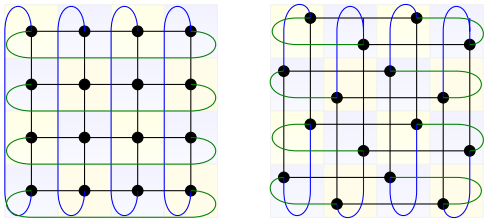
***Abstract*— Many integrated circuits now consist of multiple processing elements that can be regularly tiled across the two-dimensional surface of a die. A regular grid-based network-on-chip shares resources that simplify communication between components at the cost of increased latency, particularly under congestion. This work reports on several approaches to improve the quality of asynchronous network-on-chip design, applied to arrayed communication networks. First, we investigate the effect of replacing long "wrap" lines of a torus with transmission lines and the advantages that this method contributes to such designs. A method of deadlock free routing on torus networks using virtual channels is proven and implemented in a 65nm technology process. Finally, we evaluate the merits of these designs using a highly accurate Verilog simulation model to generate performance and power results.**

I. Introduction

Process scaling supports the implementation of many processing cores on a single planar die. Distributed memory multiprocessors are an important class of such multiprocessor designs. As the number of cores on chips has increased, power has become a primary design constraint. Clockless circuits provide event-driven computation and have been shown to provide high performance with low power overhead, which makes them an ideal candidate for network on-chip (NoC) communication. These attributes have lead clockless circuits to be used in a number of NoC applications, in both traditional NoC applications [1] and the emerging field of neuromorphic computing [2].

Traditional NoCs consist of arrays of a local microprocessor and associated memory, which communicate via a mesh-based network-on-chip (NoC) communication fabric. The size of the array is defined by n, the number of processors on each edge of the array. The planar nature of the die maps well to regularly spaced two dimensional processing and NoC arrays such as a square 8×8 array containing 64 processor and memory banks. Each node communicates with its four adjacent nodes in the array.

The hop count is the number of routers a message must traverse when communicating between source and destination nodes. The worst case hop count for a square planar mesh unconnected at the periphery is $2(n-1)$. The edges of an array can be connected by adding *wrap* wires, as shown in Fig. 1. This creates a three dimensional interconnect where the wrap wires create a toroidal NoC structure. Adding wrap lines doubles the network bisection, but also reduces the worst case hop count of a mesh by a factor of two to $2\lfloor n/2 \rfloor$.



(a) Wrapped torus (b) Folded torus

Fig. 1: Mesh based routing examples

While 3D torus routers reduce the worst case hop count, they introduce several complications. First, torus wrap lines invalidate the deadlock free property of common mesh based XY routing algorithms. Cycles are formed in the X and Y planes that don't exist in a mesh; these cycles produce deadlock scenarios. Additionally, wrap lines are difficult to map onto a two dimensional plane. Simply connecting the edges of a planar array of nodes as shown in Fig. 1(a) results in short local interconnect between nodes on the plane, but long wrap wires that are $n-1$ times longer than the in-plane connections. This results in two very different delay values depending on which type of interconnect is part of the path, be it local or wrap lines. Additionally long wrap lines significantly penalize clockless circuits, which rely on handshake protocols across the long links. One solution is to create a "folded" torus that is mapped to the planar chip, shown in Fig. 1(b). This creates communication links that are all similar in length, delay, and energy by penalizing the wire delay of the short links.

This paper directly addresses the performance and design issues that arise in toroidal interconnection networks. A method of overcoming toroidal deadlock scenarios is presented, which employs virtual channels. The deadlock freedom of this method and pipeline constraints of virtual channel logic are proven. A new approach to designing torus networks is presented, which demonstrates improvements in energy and performance efficiency. This style employs two very different types of interconnect: diffusive wires and transmission lines.

This work employs carefully designed high performance, low power on chip transmission lines to improve the power and performance of network-on-chip applications. The mesh based routing core, with short interconnects between each core, is implemented with traditional diffusive wires. Long wrap lines are implemented with transmission lines, which

begin to gain power and performance advantages above 2 mm of wire length. We call our design a "transmission line enabled clockless network on chip" (TECNO).

A flexible simulation model has been designed which generates standard cell library gate accurate energy and performance results. The model is based on Verilog simulations of hardware under evaluation. The design uses a cell-based design methodology for the routers and a spice based model for wire delay and energy. The test bench reported in this work runs simulated traffic on an 8×8 tiled system consisting of 64 nodes.

II. Background and Related Work

A. Low Diameter and Asynchronous Networks on Chip

Contemporary NoC designs prefer mesh topologies due to their simplicity, ease of physical mapping, and use of short wires [1]. These topologies have a strong correlation between logical minimality and physical proximity, making them perfectly suitable for applications with mostly local traffic. However, they suffer from having a larger hop count, latency, and lower path diversity. Designers have tried to solve this problem by using high radix routers to build complicated topologies such as flattened butterfly, folded CLOS, and dragonfly [3]. While these hierarchical topologies work very well as the underlying networks in high performance supercomputers, they are expensive to implement in a power constrained on-chip system due to the complexity associated with the routers. The torus offers advantages from both domains. It maintains the design and routing simplicity of meshes, while providing connectivity between nodes at the extremities. Compared to the mesh, tori offer twice the channel bisection, half the channel load for uniform and worst case traffic, and 24% lower hop count [3].

B. Asynchronous NoC Design

There are a number of examples of clockless interconnection network designs. One approach develops a NoC architecture that provides connection oriented guaranteed service and connection-less best effort routing [4]. Router designs using virtual channels for multiple service levels have been demonstrated [5], [6], [7]. Dynamic virtual channel allocation to provide improved best effort service with reduced area overhead has also been studied [7]. Three-port lightweight routers for binary fan-in, binary fan-out Mesh-of-Trees (MoT) networks are presented in [8]. Another approach adds limited dynamic reconfiguration capabilities to the network by identifying critical routes and bypassing arbitration on them [9]. An early arbitration resolution using a lightweight monitoring system for MoT networks is proposed in [10]. A 2-phase bundled data design for a 5-port router suitable for mesh networks have been implemented [11]. The lightweight monitoring idea for early arbitration and channel allocation is applied to the 5-port router design in [12]. Asynchronous channels have been combined with synchronous routers to enable multi-hop channel reservation and router bypassing [13]. A hybrid design of 4-phase bundled data routers with 2-phase channels is offered in [14]. Recently, Intel demonstrated a fabricated hybrid circuit/packet switched NoC design that employs handshake-based communication [1]. A time division multiplexed (TDM) asynchronous NoC architecture is presented that allows message passing among processor nodes in a multiprocessor environment [15]. However, none of these designs present a method of non-blocking handshake driven arbitration that enables deadlock-free routing in tori configurations.

TABLE I: Repeated Diffusive Wire Evaluation

Wire (metal layer)	distance (mm)	Latency (ps)	Energy (pJ/bit)	Repeater Size (μm)
RC1 (M1)	7	963.8	1.08	3.6
RC2 (M5)	7	775.5	0.78	3.6
RC1 (M1)	1	137.1	0.154	3.6
RC2 (M5)	1	110.8	0.111	3.6

C. Global Signaling using Transmission Lines

The high latency and energy costs associated with conventional repeated RC interconnects have prompted researchers to investigate alternative physical transport media for longer channels. Low swing equalized wires, silicon photonics, 3D stacking, and wireless communication are some examples for this. However, many of them suffer from some glaring drawbacks, such as noise immunity, energy expense, difficulty in fabrication, time-to-market and large chip area. An excellent resource comparing the relative merits of alternative transport media is provided [16].

There are a number of design choices while implementing a transmission line based communication system. The topology of the signal and return paths offer the options of coplanar or microstrip configuration. Coplanar lines provide good shielding from neighboring wires, but the wiring pitch is high due to the presence of guard bands. On the other hand, microstrip lines offer a close correlation between the signal and return paths, but they have the disadvantage of running parallel lines in consecutive metal layers, which is prohibitive in a regular CMOS process. In this work, we employ coplanar transmission lines as long range interconnects. Depending on the signaling scheme, the transmission line (TL) system can be single ended or differential. Single ended lines have a lower wiring pitch as compared to differential lines; on the other hand differential signaling offers many advantages such as self-reference, better noise rejection, increased signal swing, subtractive effect of mutual inductance, etc. Based on this, we deem it as the better of the two alternatives. Various on chip TL designs have been demonstrated [17], [18], [19], [20], [21]. Replacing long, diffusive wires of low diameter networks with equalized, higher average latency interconnects has been studied previously [22]. Similarly, our design maintains the regularity of a torus, using 5-port routers, but equalizes delay by replacing long diffusive wires with low latency transmission lines.

III. Approach

Designs in this paper implement a 64 core architecture where each core utilizes 1 mm^2 of silicon area. Center-to-

center distance between adjacent cores in the wrapped torus is 1 mm, with a 7 mm distance between the farthest cores on the X or Y axis. All links on the folded torus topology are 2 mm in length. Three different 64-core toroidal network designs are compared: a wrapped torus, a folded torus, and a transmission line enabled wrapped torus design (TECNO) where the wrap lines use serializer-deserializer (SERDES) transmission line communication.

We have developed a highly accurate and flexible Verilog NoC simulation platform (Sec. VII). The router implementation is accurately simulated for performance and power with physical design information back-annotated into the Verilog simulator. The wire interconnect for both TL and diffusive links are emulated by Verilog modules, which specify an appropriate fixed delay for each link employing worst-case values. The NoC simulation platform runs simulated traffic on the network architecture while recording all signal transitions in a value charge dump (vcd) file.

Wire properties for both the diffusive and transmission line links have been accurately simulated and evaluated in SPICE and other low-level models. Results are shown in Table I. All circuits were designed with regular Vt static Artisan cell library on the IBM10sf process. Transmission line simulations include full variation analysis and eye diagram calculations. Diffusive wire values are modeled under typical process conditions with a derating factor applied to account for variations including process, coupling, and handshake setup times using equations from [23].

Architecture performance is derived from the Verilog simulation. Router power is derived from PrimeTime by back annotating delays, parasitics, and node activity from the vcd file. To calculate interconnect power, we have written a program to extract interconnect transition counts from the vcd file and multiply those results by energy per transition derived from SPICE wire models.

IV. Transmission Line Communication System

Unlike most other disruptive low latency signaling mechanisms, transmission lines can be realized using industry standard CMOS processes and standard back-end-of-line (BEOL) stack. To engineer on-die transmission line behavior, the wire resistance needs to be suppressed and the inductive component boosted. This can be achieved by using wider wires on thick metal layers and correctly designing signal and return paths. High speed circuits are designed to send high frequency pulses along the lines. At the receiving end, a sense amplifier restores the signal to its original level. To utilize the full bandwidth potential of the lines, a SERDES apparatus is deployed. This approach decreases bandwidth, but also mitigates the pitch overhead of transmission lines due to wider wires and return paths.

A. SPICE Simulation using the Field Solver

On chip transmission lines require careful electromagnetic analysis, modeling, and design. Transmission line interconnect design needs a mixed signal approach in addition to the standard digital CAD tool flow. Since the signal is being propagated in the form of an electromagnetic wave, field solver simulations are performed. The top metal layers of the BEOL are geometrically described in the Star HSPICE 2D field solver with accurate metal and dielectric properties and wire dimensions. Various configurations are modeled and their RLGC matrices are extracted. These values are used to define the transmission lines for transient simulations using the RLGC model. Slight modifications in the wire dimensions cause a discernible change in their properties.

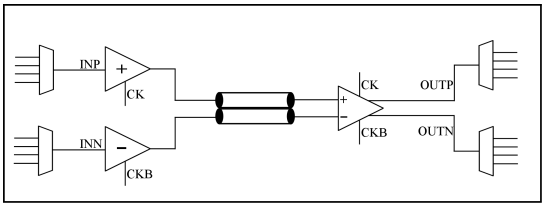


Fig. 2: TL Communication System

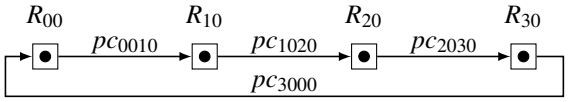
TABLE II: TL Evaluation for 7 mm Link

Transmission Line (opt)	Latency (ps)	Energy (pJ/bit)	Pitch (μm)	Eye Height (V)	Eye Width (ps)
TL1 (energy)	338.3	0.54	2.8	0.88	339.6
TL2 (latency)	292.9	0.60	7.6	0.85	337.9

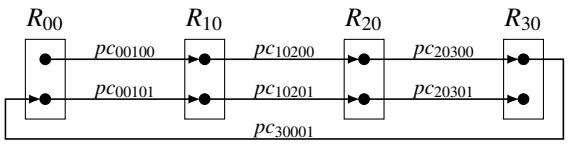
The transceiver circuitry for a TL based communication system includes a transmitter that generates full swing, high frequency data signals to drive the long channel, and a sense amplifier to reconstruct the attenuated signal at the far end of the transmission line. The primary objective in choosing transmitter and receiver designs was to allow the use of conventional static CMOS libraries for implementation. Voltage mode drivers were selected since they make receiver designs simple and provide better performance in the presence of noise. The driver circuitry consists of interleaved tristate drivers that serialize the incoming data streams and launch them into the transmission line [18]. The final stage of the driver provides the drive strength and impedance matching required to drive the long channel. The receiver circuit has a pair of comparators to detect the signal at the output of the transmission line. Use of dynamic logic facilitates high speed operation. The transmitter and receiver transistors were sized to optimize the channel performance. Multiplexers and decoders are used to build simple SERDES circuits. Fig. 2 shows the basic block diagram of the TL communication system.

B. Comparisons

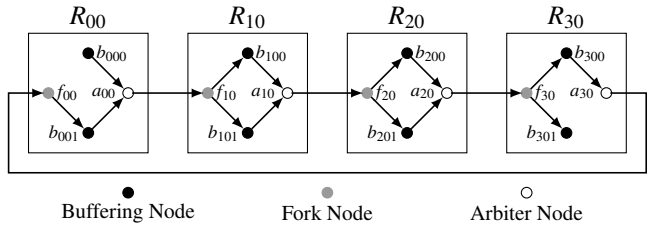
Various metrics are examined to gauge the performance of the TL communication system. Latency, energy, and noise immunity are compared for different wireline structures. The transmission lines are optimized for latency and energy. Delay calculations take into account the overhead of the transmitter-receiver pair as well as the latency along the line. Energy numbers are calculated on a per bit basis. Eye height and width are measured to gauge the link performance. All measurements are done for a 7 mm long transmission



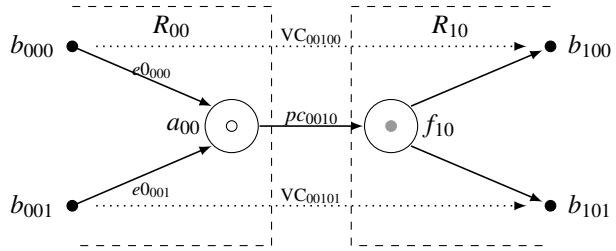
(a) Network cycle generated by a slice of the network



(b) Deadlock free implementation using two physical channels



(c) Mapping physical channel implementation to virtual channel implementation using fork and arbiter nodes



(d) A portion of Fig. 3c that specifies how buffering and non-buffering nodes are connected. The persistency property of Def. 3 is represented with the dotted arcs.

Fig. 3: Asynchronous network-on-chip structures

line using the ASU 65nm BSIM models using HSPICE [24]. Table II shows the measurement results for two TL configurations; one optimized for energy, and the other for latency. The measurements are comparable to results reported in [25], [26]. Based on the simulation results, we elect the optimal configuration for use in TECNO. For comparison, we also report the equivalent repeated diffusive wire values for 7 mm wires for M1 and M5 metal layers in Table I. A number of repeater sizes were evaluated and the optimal value is chosen. We observe that delay optimized transmission lines have 2.6× lower latency than their RC counterparts. The energy efficiency of properly designed transmission lines is 1.4× better than the diffusive wires.

V. Deadlock Properties of Asynchronous Tori

XY routing algorithms prevent deadlock in a 2D mesh by removing cycles during network packet delivery [3]. These algorithms are insufficient to prevent deadlock in toroidal meshes because each row and column contain a physical cycle due to wrap lines (Fig. 1). Fig. 3a illustrates the cycle in the horizontal dimension of a torus topology where router at column x and row y is represented as R_{xy}. The wrap line connecting R_{30} and R_{00} introduces cycles during network packet delivery which can result in deadlock dependencies.

Cycles in a toroidal mesh can be broken by introducing two physical channels in each dimension as illustrated in Fig. 3b. Each router at indices x, y contains buffers b_{xyk} that route on physical channel k.

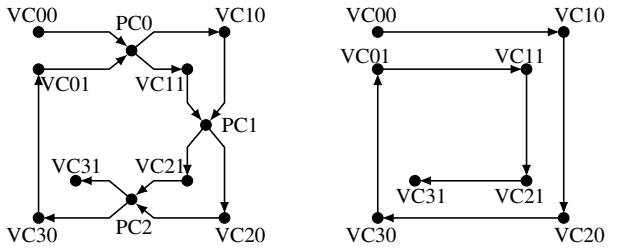
The prohibitive routing cost of independent physical channels makes this solution undesirable. Virtual channels (VC) alleviate routing costs by multiplexing VC's across a physical channel. VC segments VC_{00100} and VC_{00101} are removed and represented by the dotted lines in Fig. 3d. Data in the VCs are now routed through a shared physical channel (pc_{0010}).

Previous work proves deadlock freedom of XY higher order channel routing for clocked toroidal networks by introducing a channel dependency graph. A channel dependency graph (CDG) represents resource dependencies as a directed graph for a given interconnection network and routing function. It was proven that iff a channel dependency graph G_d is without cycles, as in Fig. 4b, it is deadlock free [27]. For clocked circuits, the dependency graph of Fig. 4a which represents a VC implementation can be proved to behave as the graph of Fig. 4b if, for each row and column, a total order on channel dependencies can be enforced giving priority to traffic on VC1 over VC0 [27].

Unfortunately proofs that employ virtual channels to create a non-cyclical channel dependency graph are insufficient to prove deadlock freedom when request-acknowledge handshaking is employed across the communication channels. The handshake protocol itself creates deadlock even with virtual channels because physical channels are not re-allocated until the handshake cycle completes. Therefore traffic priority can not be guaranteed between multiple virtual channels. Thus the cycles in G_d remain, producing deadlock.

We prove deadlock free routing in a toroidally wrapped dimension using request-acknowledge handshake communication when: (a) Forward progress of bubbles and tokens using a negative acknowledgment handshake protocol is proven. (b) Pipelining is not allowed in fork and join logic between VC buffers. (c) New communication is injected sufficiently early in the channel dependency graph. (d) Fair arbitration of the physical channel and new communication injection is employed. We then prove that a complete toroidal network is deadlock free if a deadlock free mesh routing algorithm is employed using deadlock free dimensional routing.

Definition 1: A virtual channel network is a directed graph $D = \{B, A, F, E0, E1\}$ where $B = \{b_{000}, \ldots, b_{xyk}\}$ is a set of buffering nodes where b_{xyk} represents virtual channel k of router node xy, $A = \{a_{00}, \ldots, a_{xy}\}$ is a set of fair arbitration nodes where each node a_{xy} grants mutually exclusive access to a shared physical channel upon request from multiple virtual channels, $F = \{f_{00}, \ldots, f_{xy}\}$ is a set of fork nodes which routes a request from a physical channel to destination virtual channel nodes b_{xyk}, $E0$ is a set of edges $e0_{xyk} = (b_{xyk}, a_{xy})\ \forall x, y, k$, and $E1$ is a set of a edge pairs $e1_{xyk} = ((a_{x_s y_s}, f_{x_d y_d}), (f_{x_d y_d}, b_{x_d y_d k}))\ \forall x, y, k$ between source s and destination d routers. A physical channel $pc_{x_s y_s x_d y_d}$ is the first edge of $e1$.



(a) Channel dependency graph with shared physical channels.

(b) Non-cyclical CDG resulting from prioritizing VC1 traffic.

Fig. 4: Channel Dependency Graphs

Definition 2: When a message arrives at its final destination router, the message will always be consumed which creates an empty buffer b_{xyk} in network D.

Definition 3: Virtual channels are implemented with the *persistency* property. If a physical channel in a dimension is not a wrap line (e.g. $x_d = x_{s\pm1}, y_d = y_s$) then the source virtual buffer channel k will always route to the destination virtual buffer on the same channel ($e0_{xyk_s}, e1_{xyk_d}$ where $k_s = k_d$). If a physical channel is a wrap line (e.g. $x_d > x_{s\pm1}, y_d = y_s$) then VC0 source buffers connect to VC1 destination buffers ($b_{x_s y_s 0}, b_{x_d y_d 1}$), while VC1 source buffers do not drive a physical channel.

The persistency property is illustrated with the dotted lines for unwrapped communication links in Fig. 3d. The wrap lines are connected as shown in Fig. 3c.

Theorem 1: Deadlock freedom requires that all new traffic injected into a dimension either through changing dimensions or newly injected into the router will be routed to buffers on virtual channel 0 (b_{xy0}) or at least n routers away from the final router on virtual channel 1 if the message requires n hops in that dimension.

Proof: By Def. 1 the final router does not have an $e0$ link on VC1. Thus any message that requires that link will deadlock. ■

Definition 4: The capacity function $C(X)$ returns the buffering capacity for a buffer b_{xyk}, arbiter a_{xy}, or fork f_{xy} in D. Communication between source buffer $b_{x_s y_s}$ and destination buffer $b_{x_d y_d}$ occurs using one $e0$ edge and one $e1$ edge pair.

Request-acknowledge handshake protocols will stall indefinitely until a downstream bubble allows data in a buffer to be forwarded. Once a_{xy} arbitrates between inputs and forwards the winner to f_{xy}, no other traffic can be allocated to physical channel $pc_{x_s y_s x_d y_d}$ until the current transaction completes. Thus no fixed priority across the physical channel can be enforced because requests arrive at arbitration nodes from independent sources with no common timing reference. Therefore request-acknowledge handshaking can not enforce the correct behavior of the dependency graph shown in Fig. 4b through priorities.

Theorem 2: Toroidal network D employing request-acknowledge handshaking on channels $E1$ can deadlock.

Proof: Assume the dimensional slice of a torus network in Fig 3c, where network traffic has placed a token in VC0 buffers $b_{000}, b_{100}, b_{200}, b_{300}$ and VC1 buffer b_{001}; $C(b_{xyk}) = 1\ \forall b_{xyk} \in B$, $C(a_{xy}) = 0\ \forall a_{xy} \in A$, and $C(f_{xy}) = 0\ \forall f_{xy} \in F$. If a_{00} arbitrates $e0_{000}$ over $e0_{001}$, then according to Def. 3 we get the cyclic channel dependency graph $b_{000} \rightarrow a_{00} \rightarrow f_{10} \rightarrow b_{100} \rightarrow a_{10} \rightarrow f_{20} \rightarrow b_{200} \rightarrow a_{20} \rightarrow f_{30} \rightarrow b_{300} \rightarrow a_{30} \rightarrow b_{001} \rightarrow a_{00}$. Since there are no bubbles in the cycle and due to the request-acknowledge handshaking protocol, the network is deadlocked. ■

Negative acknowledge handshaking (nack-hs) employs positive (ack) and negative (nack) acknowledgment handshakes. An ack response is returned and the token and bubble are exchanged iff the destination buffer is empty. Otherwise a nack response is returned, no transaction occurs, and the channel becomes idle. Negative acknowledge handshaking can be employed across network links [28]. Using nack handshaking, the network is deadlock free only when buffering capacity is zero for all arbiter and fork nodes.

Theorem 3: Toroidal network D employing the nack-hs protocol on edges in $E1$ can deadlock if $\exists a_{xy} \in A$ where $C(a_{xy}) > 0$ or $\exists f_{xy} \in F$ where $C(f_{xy}) > 0$

Proof: Assume the dimensional slice of a torus network in Fig 3c, and that network traffic has placed tokens in nodes $f_{10}, b_{100}, b_{200}, b_{300}, b_{001}$; $C(b_{xyk}) = 1\ \forall b_{xyk} \in B$, $C(a_{xy}) = 0\ \forall a_{xy} \in A$, $C(f_{10}) = 1$ and $\forall f_{xy} \in F, f_{xy} \neq f_{10}, C(f_{xy}) = 0$. Due to Def. 3 the cyclic channel dependency graph $f_{10} \rightarrow b_{100} \rightarrow a_{10} \rightarrow f_{20} \rightarrow b_{200} \rightarrow a_{20} \rightarrow f_{30} \rightarrow b_{300} \rightarrow a_{30} \rightarrow b_{001} \rightarrow a_{00} \rightarrow f_{10}$ exists. Assume a nack-hs protocol on $((a_{00}, f_{10}), (f_{10}, b_{100}))$. The channel dependency graph remains deadlocked even under nacking arbitration because $C(f_{10}) = 1$ and the buffer is occupied.

The proof when $\exists a_{xy} \in A$ where $C(a_{xy}) > 0$ is equivalently constructed. ■

Lemma 1: The *forward progress* property. Tokens and bubbles can always be exchanged between two adjacent routers in network D under Def. 3 using negative acknowledge handshaking across $E1$ edges if $C(a_{xy}) = 0\ \forall a_{xy} \in A$, and $C(f_{xy}) = 0\ \forall f_{xy} \in F$, and fair arbitration is employed.

Proof: Assume the router pair in Fig. 3d $\in D$ where a token exists in all buffers except b_{101}. Due to the nack-hs protocol with fair arbitration, VC1 will eventually forward the token from b_{001} to b_{101}. The same condition holds for VC0 when the bubble is in b_{100}, guaranteeing forward token and backward bubble propagation between router pairs. ■

Theorem 4: Each dimension in a torus is deadlock free if it contains the forward progress property of Lemma 1, messages are injected according to Theorem 1, and fair arbitration is used.

Proof: Virtual channels are used to create a cycle free dependency graph in each dimension according to Def. 3. According to the forward progress Lemma 1, messages can always be forwarded when a bubble exists in an adjacent router. Def. 2 ensures that tokens will be removed when reaching their destination, inserting a bubble into the network. Message injection can not deadlock the network according to Theorem 1. Fair arbitration of A ensures that each input channel can make progress. Since we have a cycle free dimension guaranteed to make forward progress, by

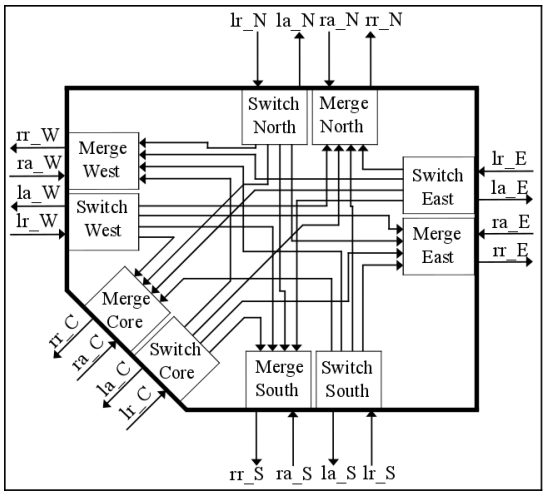


Fig. 5: Router Architecture

induction bubbles can propagate to any destination buffer in the dimension and data can propagate forward to any router in the dimension, making each toroidal dimension deadlock free. ■

Theorem 5: Toroidal network *D* is deadlock free when a deadlock free mesh routing algorithm is used where routing in each dimension is also deadlock free.

Proof: Lemma 1 ensures forward propagation of data and backward propagation of bubbles between all router pairs. Therefore, based on induction, a bubble can propagate from the end of a channel dependency graph to its beginning in any dimension slice of a torus. Thus no dimension slice will independently deadlock. If a deadlock free mesh routing algorithm is employed, no deadlocks will be created due to routing dependencies in the mesh. This results in a deadlock free design. ■

VI. Asynchronous NoC Architecture

Our network uses basic dimension ordered source routing, single-flit packets, and simple router circuits. Instead of large crossbars and buffers that are energy and area inefficient, we use simple 4:1 MUXes for switching. The two most significant bits determine the switch direction and control the data MUXes. Therefore, no additional address decoding circuitry is required. No routing logic is required because rotation of the routing bits is the only operation performed between input and output. Each packet works at flit granularity. The implementation reported in this paper contains 16 routing bits and 44 data bits. Each router has five ports, therefore enabling a kn-cube type topology. Since we use source routing, the number of routing bits depends on the network diameter. There are formulae to determine this value for regular topologies like mesh and tori. The data width of the link depends on the required throughput, power, and area budgets. The NoC does not cater to any particular transaction layer protocol, such as OCP.

A. Packet Routing

Lookup tables are used to store a set of precomputed routes for each sender-destination pair. A single route is

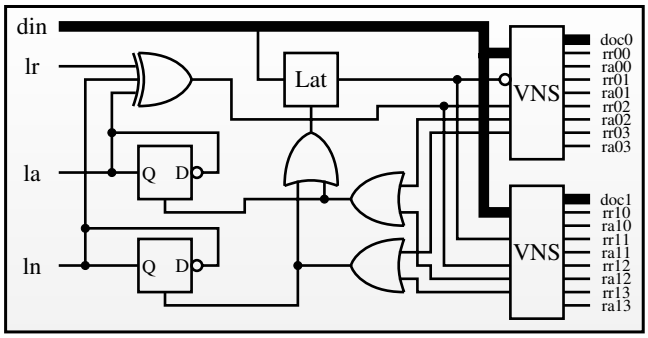


Fig. 6: Virtual Switch

calculated for each combination. The routing bits decide the direction of steering the packet at the switch module; no other computations are performed at the router. At each hop, the two most significant address bits are rotated, and the next two routing bits appear at the most significant positions. The number of routing bits depends on the maximum hop count or network diameter. No return address is required; it is directly derived from the source route. Every packet requires routing bits, thus contributing a significant overhead for large dimension networks. This means that a network that employs lower hops such as a torus immediately has an advantage over a regular mesh.

Each router has five ports which are addressed with two bits. A fixed 2-bit value is used to indicate NEWS based direction. The core port is addressed using the same direction as the port. For example, a N routing direction presented to the N input port routes the packet to the core.

B. Basic Router Design

Routers consist of virtual switch and merge blocks. The modules contain FSMs, mutual exclusion elements, non-blocking arbiters, 2-to-4 phase converters, pipeline controllers, and registers. The architecture is shown in Fig. 5.

The circuits employed by TECNO are completely asynchronous. Bundled data (BD) protocols are used throughout the design. Two phase communication is used between the routers to reduce communication overhead. The two phase channels also use a non-blocking or nacking handshake protocol. Arbitration inside the router requires four phase protocols due to the mutual exclusion (MUTEX) elements employed. Circuits for the translation between the protocols are included as part of the router design. The 2-to-4 phase boundary was placed at the periphery of the routers. All internal communication in the routers and communication with the core use four phase BD protocols.

C. Switch Module

The switch module accepts inbound packets and routes them to the appropriate destination port based on flit header address bits. The virtual switch in Fig. 6 consists of a nacking 2-to-4 phase converter and two 4-port virtual nacking switches (Fig. 7). The nacking arbiter (NARB) controller synchronizes between a nacking input (left) channel and a traditional output (right) channel. It contains a state machine

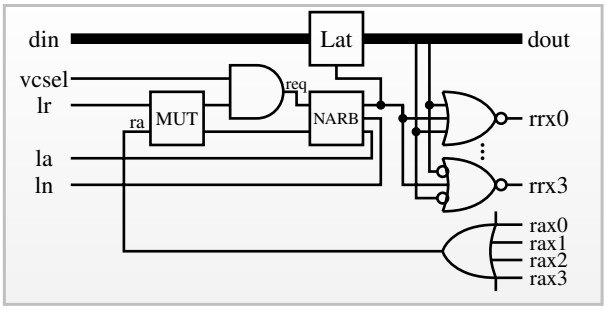


Fig. 7: Virtual nacking switch design

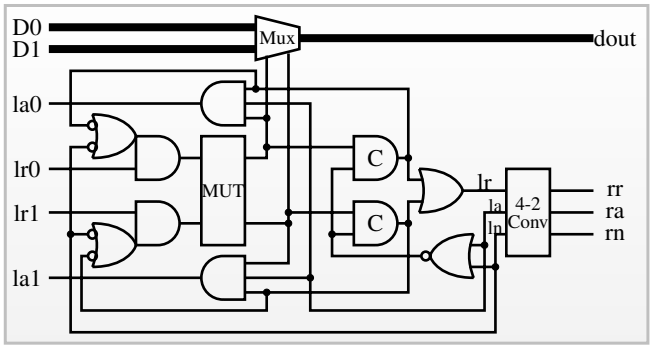


Fig. 8: Virtual Merge Circuit

that returns a nack (ln) when it receives a req (lr) and the latch is occupied; otherwise the req is acknowledged (la). The lowering of the output ack (ra) indicates the downstream channel is free. A race condition exists between a raising lr event and a lowering ra event, which move to different states in the NARBFSM. The MUTEX element ensures that these events sequenced such that the NARBFSM has no race conditions.

D. Merge Module

A 2-port version of the merge module, shown in Fig. 8, arbitrates between incoming traffic streams and forwards data to an adjacent router. The virtual merge arbitrates between two traditional input channels and converts the signals to a the single nacking physical output channel. Each merge module operates as an instance $a_{xy} \in A$ from Fig. 3c, and therefore, by Theorem 3, must be unpipelined. As a result, the path through the merge module becomes the critical path of the network.

The module contains an arbiter and a nacking 4-to-2 phase converter interface. If the destination router VC is free, it stores the data and responds with an ack (ra), which is passed back to corresponding input channel through la0 or la1. If the destination router VC buffer is full, the transaction is nack'ed (rn), which de-asserts the corresponding request through the internal path from ln to the MUTEX. This creates a 4-cycle nack handshake between lr and ln. If a request is pending on both channels (lr0 and lr1 are both asserted), then the ln handshake allows fair arbitration of VC0 and VC1 traffic through the MUTEX.

This module has additional logic to set or clear the virtual channel bit when data switches dimensions or when it crosses a wrap line.

E. Virtual Channel Overhead

The performance overhead of adding virtual channels is shown in Tab. III. Router performance was evaluated using post layout back annotated Verilog simulations. Individual router modules were inserted into a test bench to add and remove data tokens, with only performance limitations introduced by relative timing constraints of the module. Forward latency represents the delay of a data token to pass through a switch and merge module of the router. The backward latency represents the delay of the absence of data or a bubble to pass backwards through a merge and switch module of the router. The cycle time represents the delay for one handshake cycle to complete or a data token to enter the router.

TABLE III: Router Performance with and without Virtual Channels

Module	Buffer Depth	Forward Lat. (ps)	Backward Lat. (ps)	Cycle Time (ps)
VC Router	2	1185	1470	1315
Router	2	1095	735	1055

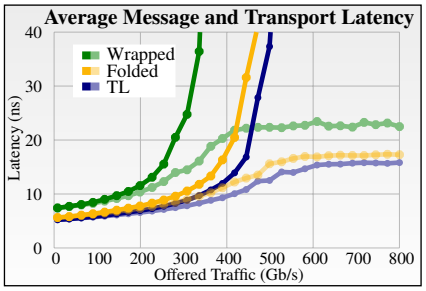
VII. Simulator

Current network simulators target cycle-accurate performance evaluation while providing customization for network topology, and scalability for large number of nodes [14], [29], [30]. Routers and channels use model abstractions, improving flexibility and run time at the cost of sacrificing accuracy. Thus this approach often lacks fidelity for RTL accurate temporal reporting and network power evaluation. While analyzing NoC architectures that employ complex transceiver and router circuits and unorthodox transmission line links, accuracy becomes the prime consideration for the simulator. In this work, we built a gate-accurate NoC simulation environment that uses Verilog simulation for network modeling and a C++ wrapper for traffic generation and performance analysis. We bridge the two domains using a SystemVerilog domain programming interface (DPI).

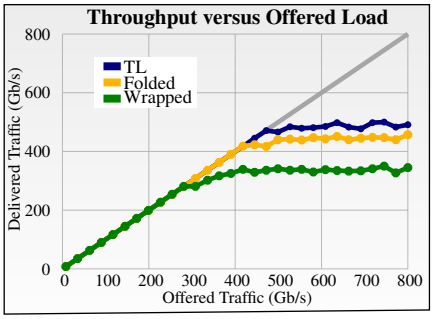
A. Verilog Environment

Network hardware simulation is performed by a Verilog simulator which allows the network simulator to report highly accurate performance and power results. The simulation environment also can display specific waveforms and behavior characteristics (such as arbitration) that are challenging or of specific interest to design quality.

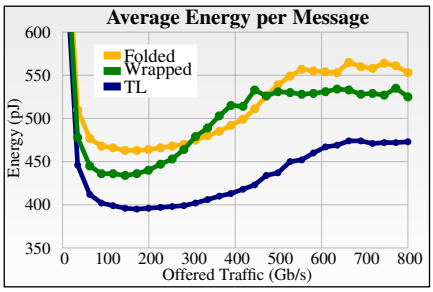
Our network fabric becomes the device under test (DUT) for the simulator. Our specific design fabric consists of fully implemented routers for each differing design configuration using the standard VLSI tool flow. Circuits are specified in Verilog, synthesized using Design Compiler, and placed and routed using SoC Encounter. ModelSim is used for functional verification, using back-annotated parasitic sdf values. PrimeTime PX is used to generate power numbers using the value



(a) Comparative Latency. Message latency includes time in delivery queues. Transport latency reports NoC delay. Lighter colors graph transport latency values.



(b) Comparative Throughput. Delivered bandwidth saturates based on network design.



(c) Comparative energy. This plots the average energy per message delivered through the network.

Fig. 9: Result Graphs

charge dump (vcd) and standard parasitic exchange format (SPEF) files from the simulations. Characterized routers are arrayed and interconnected with emulated links characterized by SPICE simulations. Values for wire delays and energies of RC and transmission line (TL) links in the 65 nm node are shown in Tab. I and II. Emulation provides additional flexibility to our simulator as we can easily switch between different wire models. The links are emulated in our network fabric as Verilog delay statements on each wire of the link based on the values in the above tables.

B. C++ Environment

In order to provide flexibility for multiple traffic patterns, simulation workloads, source-routing algorithms, and network topology, a modular simulation engine was developed. Network and environment interaction is performed through a DPI socket interface. The Verilog environment requests to add or remove message flits whenever a flit exits the network or an input handshake has completed. Each message flit is tracked through the network with unique identifier payload, which is appended with additional bits to achieve a user specified activity factor. Network performance is evaluated by time stamps of when the flit enters and leaves both the queues and network. The Verilog environment generates gate-accurate, real workload data, which provides a rich set of evaluation metrics. The traffic analyzer evaluates message transmission and latency, but also can evaluate metrics such as link utilization, energy, and hop to hop latency. This provides critical insight to network designers. The traffic analyzer uses both message queue and vcd data to evaluate network metrics.

C. Performance Evaluation

The simulations used in this paper are configured as follows. For the uniform random traffic pattern, each simulation has a fixed injection period, at which point a new message is injected using uniform random message distribution. Messages are configured to contain 10 flits. Injection period start and end values IP_s and IP_e are provided, along with the number of simulations N. Warm up and evaluation times T_w and T_e are also provided. The injection period for the simulation in terms of seconds per byte $IP_{s/b}$ is calculated as $IP_{s/b} = 1/(IP_s + i((IP_e - IP_s)/N))$. This value is used to calculate the number of messages in the warm up as $M_w = T_w/(IP_{s/b} \times M_{sz})$ where M_{sz} is the number of bytes in a message. The number of messages in the evaluation is calculated as $M_e = T_e/(IP_{s/b} \times M_{sz})$.

VIII. Evaluation

Simulations that characterize the power and performance are reported for three 8×8 toroidal NoC designs: wrapped, folded, and wrapped with transmission lines for long interconnects. Each design is simulated under uniform random traffic patterns with aggregate load data ranging from 8 Gb/s to 800 Gb/s bandwidth. This traffic pattern is instructive in evaluating various properties of a regular mesh network, as is pointed out below. Shortest hop count paths are selected for delivery between each pair of nodes. The network has a total of 128 ($2n^2$) links. Most of the links (112 or $2n(n-1)$) connect adjacent cores in the mesh topology with a 1 mm separation; there are 16 ($2n$) 7 mm wrap lines implemented as repeated wires or transmission lines. All 128 links of the folded torus have a 2 mm distance.

Average message latency and average message network transmission time are graphed in Fig. 9a. Network transmission time is the delay from the time a message enters the network until the time that it leaves the network. Differences are clear between the three topologies. Extra delay across the wrap lines provides substantial reduction in overall performance as it becomes a location for congestion. The design with diffusive wrap lines is by far the worst performer across all load cases, having a 42% larger average message latency at the lowest load value measured (8 Gb/s) when compared to the TECNO design. The transmission line design also delivers messages 8% faster on average than the folded torus, which has uniform latency on all links and no points of congestion, but larger latency across all links.

Average message network transmission time identifies the maximum sustainable bandwidth that a network can support for a given traffic pattern. This value diverges from the average message latency when input message queue time impacts total message delivery time. Transmission time increases up to the point where the network reaches maximum throughput. At this point, transmission latency remains relatively

constant. Transmission latency is another representation of overall network bandwidth, shown in Fig. 9b. This graph plots offered bandwidth versus delivered bandwidth for the three designs. It shows the injection bandwidth where the network saturates and cannot keep up with the offered traffic. The wrapped, folded, and TECNO designs saturate at approximately 337, 445, and 487 Gb/s respectively. Thus the TECNO design provides a 45% and 9% greater saturated bandwidth than the wrapped or folded designs. Note that the point where the offered versus delivered bandwidths diverge is also the point where message latency explodes (308 Gb/s and 24.8 ns, 445 Gb/s and 31.6 ns, and 499 Gb/s and 37.3 ns).

Fig. 9c shows the average energy per message plotted across the offered bandwidth. Average energy per message is initially high, as the substantial leakage components of this technology node is divided among the few messages transmitted during the simulation window. Average energy decreases until leakage becomes a small percentage of overall delivery energy at higher bandwidths. The routers are not power gated when idle. As congestion increases, so does average energy per message. Each time a message transmission across the physical link between routers cannot be stored in the destination router, the nacking channel aborts and immediately attempts to send another message. If both virtual channels have data, then the new data for the other channel is put on the bus and delivery is attempted. This increase in energy also flattens out as total network throughput saturates. B-model results were also obtained for each of the networks. Due to volume fluctuations of the B-model, the networks saturate at slightly lower injection rates, but show similar latency and power relationships.

The transmission line design demonstrates significantly better energy per message than the other networks. The folded design in general offers the worst energy efficiency since there is more average wire capacitance switched per message. For moderate throughput conditions (172 Gb/s offered bandwidth) the minimum average energy per message for the wrapped and folded design is 10% and 17% greater than TECNO.

IX. Conclusions

This paper presents details on the design of a high performance asynchronous router which uses negative acknowledge handshaking and virtual channels to prevent deadlock in a torus connected mesh. Proofs of deadlock freedom and resultant physical channel pipeline restrictions are presented. In order to guarantee virtual channel allocation and deadlock freedom it was proven that merge and arbitration logic cannot be pipelined, which significantly reduces virtual channel router performance. Transmission lines have been shown to provide reduced average latency and power over traditional long wrap lines, with uniform interconnect delay of a folded torus. Unfortunately, channel arbitration logic, which cannot be optimized through pipelining from Theorem 2, forms the critical path of the router circuit and limits performance gained from adding transmission lines. Future work should implement protocols which can further leverage these benefits such as those that ensure availability of downstream resources [26]. A configurable simulator was developed to evaluate network performance and traffic. The simulator is used to compare performance and power for three torus designs implementing 64 core architectures: wrapped, wrapped with transmission lines, and folded designs. The transmission line torus shows superior latency (42%, 8% faster), aggregate NoC bandwidth (45%, 9% greater bandwidth), and energy per message (10%,17% lower energy) than the wrapped or folded designs respectively.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1111533. The authors would like to thank Matthias Füegger for his contributions and guidance during the revision of this publication. Kenneth S. Stevens declares financial interest in Granite Mountain Technologies, Inc. which commercializes high performance low power integrated circuits.

References

[1] G. Chen, M. A. Anders, H. Kaul, S. K. Satpathy, S. K. Mathew, S. K. Hsu, A. Agarwal, R. K. Krishnamurthy, V. De, and S. Borkar, "A 340 mV-to-0.9 V 20.2 Tb/s Source-Synchronous Hybrid Packet/Circuit-Switched 16 x 16 Network-on-Chip in 22 nm Tri-Gate CMOS," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 1, pp. 59–67, Jan 2015.

[2] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, G.-J. Nam, B. Taba, M. Beakes, B. Brezzo, J. B. Kuang, R. Manohar, W. P. Risk, B. Jackson, and D. S. Modha, "TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 10, pp. 1537–1557, Oct 2015.

[3] W. J. Dally and B. P. Towles, *Principles and Practices of Interconnection Networks*. Elsevier, 2004.

[4] T. Bjerregaard and J. Sparsø, "Implementation of guaranteed services in the MANGO clockless network-on-chip," *IEEE Proceedings on Computers and Digital Techniques*, vol. 153, no. 4, pp. 217–229, July 2006.

[5] D. R. Rostislav, V. Vishnyakov, E. Friedman, and R. Ginosar, "An Asynchronous Router for Multiple Service Levels Networks on Chip," in *Proceedings. 11th IEEE International Symposium on Asynchronous Circuits and Systems, 2005. ASYNC 2005.*, March 2005, pp. 44–53.

[6] R. R. Dobkin, R. Ginosar, and A. Kolodny, "QNoC asynchronous router," *Integration*, vol. 42, no. 2, pp. 103–115, February 2009.

[7] T. Feliciian and S. B. Furber, "An asynchronous on-chip network router with quality-of-service (QoS) support," in *IEEE International SOC Conference, 2004. Proceedings.*, Sep. 2004, pp. 274–277.

[8] M. N. Horak, S. M. Nowick, M. Carlberg, and U. Vishkin, "A Low-Overhead Asynchronous Interconnection Network for GALS Chip Multiprocessors," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 4, pp. 494–507, April 2011.

[9] G. Gill, S. S. Attarde, G. Lacourba, and S. M. Nowick, "A Low-Latency Adaptive Asynchronous Interconnection Network Using Bi-Modal Router Nodes," in *Fifth IEEE/ACM International Symposium on Networks on Chip (NoCS)*, May 2011, pp. 193–200.

[10] G. Faldamis, W. Jiang, G. Gill, and S. M. Nowick, "A low-latency asynchronous interconnection network with early arbitration resolution," in *19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan 2014, pp. 329–336.

[11] A. Ghiribaldi, D. Bertozzi, and S. M. Nowick, "A Transition-Signaling Bundled Data NoC switch Architecture for Cost-Effective GALS Multicore systems," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, March 2013, pp. 332–337.

[12] W. Jiang, K. Bhardwaj, G. Lacourba, and S. M. Nowick, "A lightweight early arbitration method for low-latency asynchronous 2D-mesh NoC's," in *52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2015, pp. 1–6.

[13] T. Krishna, C.-H. O. Chen, W.-C. Kwon, and L.-S. Peh, "Smart: Single-Cycle Multihop Traversals over a Shared Network on Chip," *IEEE Micro*, vol. 34, no. 3, pp. 43–56, May 2014.

[14] D. Gebhardt, J. You, and K. S. Stevens, "Design of an Energy-Efficient Asynchronous NoC and its Optimization Tools for Heterogeneous SoCs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 9, pp. 1387–1399, Sept. 2011.

[15] E. Kasapaki and J. Sparsø, "Argo: A Time-Elastic Time-Division-Multiplexed NOC using Asynchronous Routers," in *2014 20th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, May 2014, pp. 45–52.

[16] A. Karkar, T. Mak, K.-F. Tong, and A. Yakovlev, "A Survey of Emerging Interconnects for On-Chip Efficient Multicast and Broadcast in Many-Cores," *IEEE Circuits and Systems Magazine*, vol. 16, no. 1, pp. 58–72, 2016.

[17] H. Ito, M. Kimura, K. Miyashita, T. Ishii, K. Okada, and K. Masu, "A Bidirectional- and Multi-Drop-Transmission-Line Interconnect for Multipoint-to-Multipoint On-Chip Communications," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 4, pp. 1020–1029, April 2008.

[18] H. G. Rhew, M. P. Flynn, and J. Park, "A 22Gb/s, 10mm On-Chip Serial Link over Lossy Transmission Line with Resistive Termination," in *2012 Proceedings of the ESSCIRC*, Sept 2012, pp. 233–236.

[19] A. Carpenter, J. Hu, J. Xu, M. Huang, H. Wu, and P. Liu, "Using Transmission Lines for Global On-Chip Communication," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 2, pp. 183–193, June 2012.

[20] Q. Hu, P. Liu, M. C. Huang, and X.-H. Xie, "Exploiting Transmission Lines on Heterogeneous Networks-on-Chip to Improve the Adaptivity and Efficiency of Cache Coherence," in *Proceedings of the 9th International Symposium on Networks-on-Chip*, ser. NOCS. ACM, 2015.

[21] Y. Zhang, R. Dobkin, A. Unikovski, D. Nahmanny, G. Samuel, M. Moyal, and R. Ginosar, "A 1.4×FO4 self-clocked asynchronous serial link in 0.18 μm for intrachip communication," *Integration*, vol. 59, pp. 190–197, 2017.

[22] A. Joshi, B. Kim, and V. Stojanović, "Designing Energy-Efficient Low-Diameter On-Chip Networks with Equalized Interconnects," in *17th IEEE Symposium on High Performance Interconnects (HOTI)*, Aug 2009, pp. 3–12.

[23] K. S. Stevens, P. Golani, and P. A. Beerel, "Energy and Performance Models for Synchronous and Asynchronous Communication," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 3, pp. 369–392, March 2011.

[24] Y. Cao, T. Sato, M. Orshansky, D. Sylvester, and C. Hu, "New Paradigm of Predictive MOSFET and Interconnect Modeling for Early Circuit Simulation," in *Proceedings of the IEEE 2000 Custom Integrated Circuits Conference*, May 2000, pp. 201–204.

[25] M. F. Chang, J. Cong, A. Kaplan, M. Naik, G. Reinman, E. Socher, and S.-W. Tam, "CMP Network-on-Chip Overlaid With Multi-Band RF-Interconnect," in *14th International Symposium on High Performance Computer Architecture*, Feb 2008, pp. 191–202.

[26] S. Das, G. Manetas, K. S. Stevens, and R. Suaya, "Leveraging the geometric properties of on-chip transmission line structures to improve interconnect performance: A case study in 65nm," in *2018 IEEE Seventh International Symposium on Networks-on-Chip (NoCS)*, April 2013, pp. 1–2.

[27] W. J. Dally and C. L. Seitz, "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks," *IEEE Transactions on Computers*, vol. C-36, no. 5, pp. 547–553, May 1987.

[28] K. S. Stevens, S. V. Robison, and A. L. Davis, "The Post Office – Communication Support for Distributed Ensemble Architectures," in *Proceedings of 6th International Conference on Distributed Computing Systems*, May 1986, pp. 160–166, best paper award.

[29] F. Fazzino, M. Palesi, and D. Patti, "Noxim: Network-on-chip simulator," *URL: http://sourceforge. net/projects/noxim*, 2008.

[30] N. Jiang, D. U. Becker, G. Michelogiannakis, J. Balfour, B. Towles, D. Shaw, J. Kim, and W. J. Dally, "A Detailed and Flexible Cycle-Accurate Network-on-Chip Simulator," in *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, April 2013, pp. 86–96.