

An Area-Efficient High-Throughput Hybrid Interconnection Network for Single-Chip Parallel Processing

Aydin O. Balkan

University of Maryland, Department of Electrical and Computer Engineering

Gang Qu

Uzi Vishkin

{balkanay, gangqu, vishkin}@umd.edu

ABSTRACT

Single-chip parallel processing requires high bandwidth between processors and on-chip memory modules. A recently proposed Mesh-of-Trees (MoT) network provides high throughput and low latency at relatively high area cost. In this paper, we introduce a hybrid MoT-BF network that combines MoT network with the area efficient butterfly network. We prove that the hybrid network reduces MoT network's area cost. Cycle-accurate simulation and post-layout results all show that significant area reduction can be achieved with negligible performance degradation, when operating at same clock rate.

Categories and Subject Descriptors

B.4.3 [Interconnections (Subsystems)]: Topology; C.2.1 [Network Architecture and Design]: Packet-switching Networks

General Terms

Design, Performance

Keywords

On-chip networks, Mesh-of-Trees, Hybrid networks

1. INTRODUCTION

Some easy-to-program parallel processing approaches require high memory bandwidth to satisfy concurrent memory requests from multiple processors on the same chip. Usually, the global memory space is partitioned into multiple smaller modules to allow concurrent accesses. An on-chip network is required to interconnect parallel processors and multiple memory modules, and convey memory requests and data between them. Studies have shown that traditional bus-based networks will not be able to provide sufficient bandwidth [2], and Network-on-Chip (NoC) architectures will soon replace them as the number of on-chip processors increases rapidly.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2008, June 8–13, 2008, Anaheim, California, USA.

Copyright 2008 ACM ACM 978-1-60558-115-6/08/0006 ...\$5.00.

Earlier studies [5] proposed a specific Mesh-of-Trees (MoT) on-chip network that provides high performance (high throughput and low latency) for large amounts of parallelism with high traffic rates. On average, MoT provides a throughput up to 0.98 flits per cycle on a 64-terminal network ($2Tbps$ cumulative throughput with a $1GHz$ hypothetical clock and 32-bit flits), much higher than butterfly and hypercube. However, the register area of MoT grows quadratically with number of network terminals, making it impractical for large systems with many terminals.

In this study, we propose hybrid MoT networks called MoT-BF, where we replace part of MoT network by butterfly (BF) networks of small scale. A BF network is area efficient, but it performs poorly under heavy traffic in terms of throughput and latency, particularly when the number of network terminals is large. In a hybrid network, traffic is diluted through MoT network; hence each mini-BF is subject to low traffic, mitigating the high traffic performance loss of pure BF network. We conduct a comprehensive evaluation of the proposed hybrid MoT-BF network in terms of area, latency and throughput. Mathematical analysis, cycle-accurate simulation and post-layout results all show that the proposed hybrid MoT-BF network can significantly reduce the area cost of MoT network with negligible performance degradation.

2. BACKGROUND AND RELATED WORK

We first briefly describe the underlying memory model. Then, we summarize key features of the MoT network and two butterfly networks that we will use for comparison.

2.1 A Memory Architecture for Single-Chip Parallelism

Using multiple memory modules (or banks) has been a common approach to increase memory bandwidth. In general, the global memory space is partitioned over the modules, and accesses to different modules are handled concurrently. A universal hashing type approach can be used to avoid pathological access patterns [1, 3, 10]. Figure 1 depicts a *Uniform Memory Access* (UMA) type memory structure used in a recent single-chip parallel architecture, which is designed to optimize single-task completion time [14]. The globally shared memory space is partitioned into multiple memory modules, the same number as the number of processing clusters (PCs) P_i s on chip. Each memory module consists of on-chip cache and off-chip memory portions. This structure completely avoids cache coherence issues because the processors do not have their private caches [14]. How-

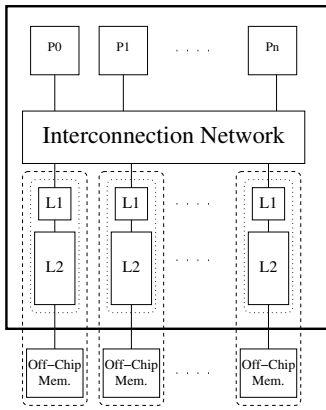


Figure 1: Global memory is partitioned into modules (separated by dashed lines). Each module has its own possibly multi-level on-chip caches (within dotted lines) [5].

ever, this structure requires the connection between each PC and each memory module. It significantly increases performance demands of the interconnection network, particularly when today’s single-chip multi-processor is pushing dozens and even hundreds of processing clusters.

2.2 Mesh of Trees Network

The MoT network is designed to provide the needed bandwidth for UMA-type memory architectures such as the one described in Section 2.1. NoC architectures that are built with 2D-Mesh topology have $O(\sqrt{N})$ bisection bandwidth [7], where N is the number of PCs. Therefore, they cannot efficiently support the expected traffic of this memory architecture. Other networks with $O(N)$ bisection bandwidth such as butterfly, hypercube and fat trees will run into switch complexity and packet contention issues, and will yield low performances [5].

Figure 2 shows the MoT topology of [5] with four PCs and four memory modules (MMs). Unlike earlier MoT topologies of [8, 12, 13], PCs and MMs are placed at the roots of the trees instead of the leaves.

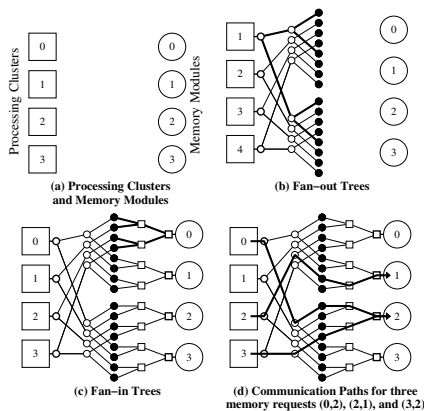


Figure 2: Mesh of Trees with 4 Processing Clusters and 4 Memory Modules [5].

The MoT network consists of two main structures: a set of fan-out trees and a set of fan-in trees Figure 2(b) shows the binary fan-out trees, where each PC is a root and connects to two children, and each child has two children of their own. The 16 leaf nodes also represent the leaf nodes in the binary

fan-in trees that have MMs as their roots (Figure 2(c)).

A MoT network that connects N PCs and N MMs has $\log N$ levels of fanout and $\log N$ levels of fanin trees. A memory request packet travels from source root to one of the leaves of the corresponding fanin tree. It passes to the leaf of a corresponding fanin tree, and travels to the root of that fanin tree to reach its destination (Figure 2(d)).

In the MoT network packets can compete with other packets from the same source while passing through fan-out trees; and with other packets to the same destination while passing through fan-in trees. It is guaranteed that, unless the memory access traffic is extremely unbalanced, packets between different sources and destinations will not interfere. Therefore, MoT network provides high average throughput that is very close to its peak throughput. Furthermore, MoT consists of less complex switches compared to other networks, and packets are routed without global scheduling. This allows higher operating frequencies and higher peak throughput. However, both fan-in and fan-out trees are $O(N^2)$ complexity and will take large area.

2.3 High-Performance Butterfly Networks

A binary butterfly network connects $N = 2^n$ terminals as shown in Figure 3(a). 16 sources and destinations are connected to each other through intermediate switch nodes. Butterfly networks have been studied previously for single-chip parallel processing [20]. In general, butterfly networks can have switches with k input and output ports ($k = 2$ in Figure 3(a)). However, switch delay increases with increasing k [15].

Registers called *virtual channels* (VCs) can be used to improve butterfly performance by increasing the amount of hardware. VCs act as buffers for incoming data packets that are stalled due to contention in later stages. A packet is stored in a VC in the switch until an output port and physical channel toward its destination becomes available [6, 7].

There are several variants of butterfly networks. One group of networks extend the regular butterfly vertically, by adding parallel resources. Extra hardware provides additional bandwidth, reduces congestion and improves throughput. Examples of this approach include *multi-butterfly* [18], *dilated butterfly* [11, 17], and *replicated butterfly* [9, 11].

Another group of networks extend the regular butterfly horizontally, by adding extra stages. This approach adds alternative paths between sources and destinations, improves traffic distribution in the network, and reduces congestion. However, without additional bandwidth throughput improvement is limited.

Advantage of replicated butterfly (RBF) over virtual-channel butterfly (VCBF) is that the logic delay of RBF switches does not increase with increased hardware. In VCBF networks, additional VCs increase logic delay [15], and may reduce system-wide clock frequency.

3. HYBRID MOT NETWORK

Earlier studies considered hybrid networks to optimize network cost and performance. A notable example is the Cube Connected Cycles (CCC) network [16], proposed to optimize high switch degree of hypercube networks. CCC network is built by replacing corners of a 3-dimensional cube with a group of terminals that are interconnected by a smaller ring network. This reduces the degree of each switch

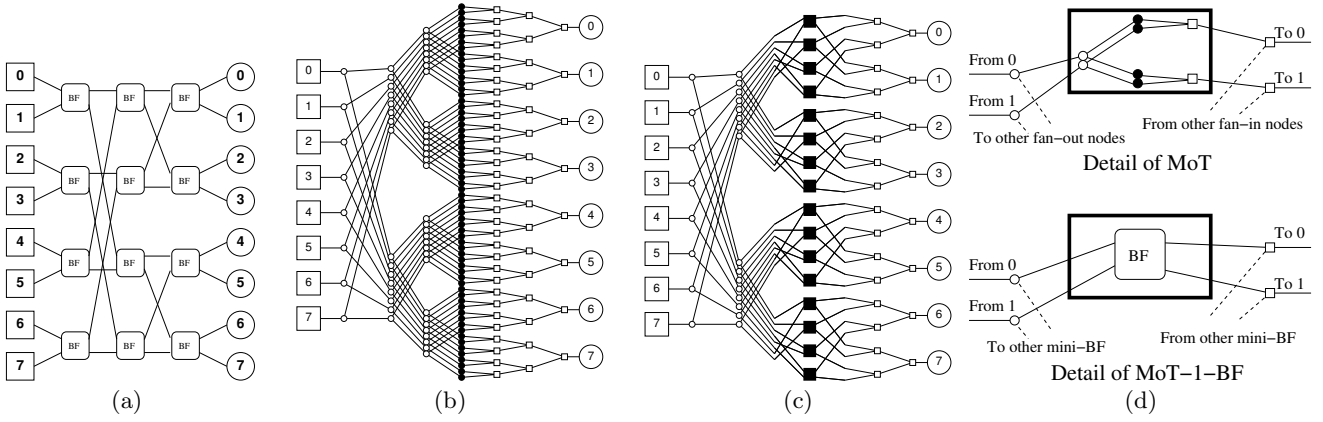


Figure 3: (a) 8-terminal butterfly network with 2×2 BF switch primitives, connecting 8 sources (numbered squares) and 8 destinations (numbered circles). (b) 8-Terminal MoT network. Small empty circles represent fan-out tree nodes, small empty squares represent fan-in tree nodes. Paths from sources $\{0,1\}$ to destinations $\{0,1\}$ are marked. (c) 8-Terminal MoT-1-BF network. 3 Innermost columns of MoT network are replaced by mini-BF networks (black squares). (d) Details of the marked paths, showing MoT (top) to MoT-1-BF (bottom) transition. BF box in (a) and (d) represents the butterfly primitive in Figure 4(d).

node of a CCC network from $O(\log N)$ to $O(1)$, where N is the number of terminals.

We propose a hybrid MoT-BF network, where inner levels of trees are replaced by mini-butterfly networks. We chose BF network due to its proven area efficiency [7].

3.1 Network Architecture

In a regular MoT network with N PCs and N MMs, we enumerate the levels of fan-out and fan-in trees by $\{0, 1, \dots, \log N - 1\}$, where the root node is at level 0, its children are at level 1, and so on. In a hybrid MoT-h-BF network, we replace the h inner levels (levels numbered $\{\log N - h, \log N - h + 1, \dots, \log N - 1\}$) of both fan-out and fan-in trees by BF networks. We refer the number $h \in \{0, 1, 2, \dots, \log N - 1, \log N\}$ to as the *hybridization level*. The remaining fan-out and fan-in trees both have $\log N - h$ levels. They are connected by $(N/2^h)^2$ mini-BF networks with h stages or 2^h terminals. (see Figures 3(b) and 3(c) for an MoT-1-BF network with $N=8$ terminals). Also note that pure MoT and pure BF networks can be represented as MoT-0-BF and MoT- $\log N$ -BF, respectively.

The main drawback of BF network is its poor performance (low throughput and high latency) at high traffic rate [5]. The proposed hybrid MoT-h-BF network reduces the traffic through mini-BF networks by the fan-out trees. Each root of the fan-out tree in the MoT-h-BF network will have $2^{\log N - h} = N/2^h$ leaves. If λ is the amount of uniform traffic in terms of flits per cycle that enters the root of fan-out tree, each input to the mini-BF networks, which is the leaf of the fan-out tree, will have a reduced traffic rate of $2^h \lambda / N$ in average, which is 0.25λ for the MoT-1-BF network with $N=8$ in Figure 3(c). This will significantly reduce the congestion and performance loss in BF networks at high traffic rates.

3.2 Switch Primitives

The original MoT tree [5] is built with three primitives (Figure 4(a, b, c)). The fan-out tree primitive performs a routing operation by directing an incoming flit to one of the two outputs. A fan-in tree primitive performs an arbitration between two incoming flits and sends the winner to the next stage. Finally, a pipeline primitive is used to cut

long wires in shorter segments if necessary. An additional butterfly primitive (Figure 4(d)) is used for building the hybrid MoT-BF network. Each primitive consists of 2 clocked b -bit¹ registers per input channel, several mux and demux and control logic that handles routing and arbitration. In an empty network, a packet spends one clock cycle in each primitive. In case of contention and stalls, proper backward signaling and using the second b -bit buffer prevents overwriting stalled data.

4. EVALUATION

We evaluate the proposed hybrid MoT-BF network in five categories, register count, minimum latency, throughput-area trade-off, network latency at different traffic rates, and post-layout throughput. We compare the proposed network to MoT, replicated BF, and virtual-channel BF networks.

4.1 Register Count

Modern VLSI processes can provide almost up to 10 metal layers, and this number increases every few generations. Earlier layout-accurate studies [4] suggest that, consequently, wire complexity becomes a secondary concern at least for reasonably small scale networks, such as 64 terminals. The network area is dominated by the data registers. Therefore, we measure register count of networks, which is directly related to the area cost.

In typical virtual-channel routing switches [7], there are v virtual channels per input and output port to improve performance. Each virtual channel uses at least one b -bit register for one data packet. In MoT, RBF and MoT-h-BF networks, each switch primitive has either one or two input and output ports and no virtual channels. In all types of switches, the control circuit consumes negligible area compared to data registers.

Mesh of Trees A MoT network consists of N fan-out and N fan-in trees, each with $(N - 1)$ nodes. The leaves do not contain switching circuits, since they are only wire connections. Using the primitive circuits of [5], the total number of b -bit registers is $R = 6N(N - 1) = O(N^2)$.

Virtual-Channel Butterfly Switches of butterfly network have a total of $2 \cdot N \log N$ input and output ports with

¹ b is the number of bits in the data path, typically 32.

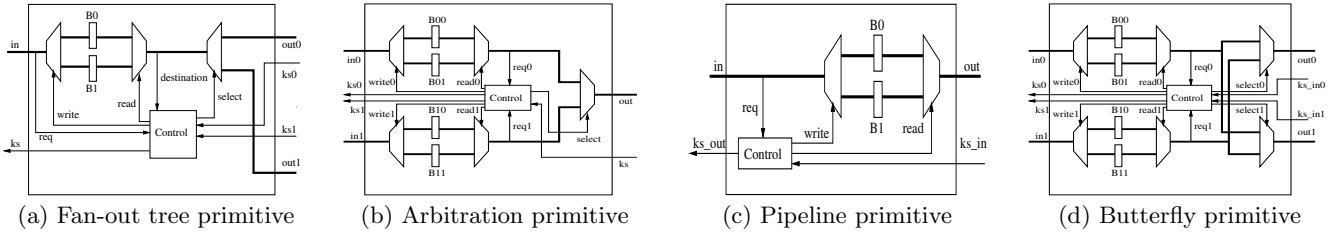


Figure 4: Switch primitives for MoT and MoT-BF networks. Data paths are marked with thick lines. Control paths are simplified. (a) Fan-Out tree primitive: One input channel, two output channels; (b) Fan-In tree (arbitration) primitive: Two input channels, one output channel; (c) Pipeline primitive: One input channel, one output channel. (d) Butterfly primitive: Two input channels, two output channels; Signals: *req*: Request; *ks*: Kill-and-Switch; *write/read*: Write and Read pointers; *B*: Storage Buffer; *select*: Result of Arbitration; *destination*: Destination address

v virtual channels each. Then, the number of registers becomes $R = 2 \cdot v \cdot N \log N = O(vN \log N)$.

Replicated Butterfly Replicated butterfly switches have two registers per input, and no virtual channels. The network consists of r copies of a regular butterfly, and binary trees between the network and source/destination modules. In total, they have $R = 6 \cdot N(r - 1) + 2 \cdot r \cdot N \log N = O(rN \log N)$ registers.

Hybrid MoT-BF A MoT-h-BF network with N terminals has N fan-out and fan-in trees, with $\log N - h$ levels. Additionally, there are $(N/2^h)^2$ mini-BF networks with h stages. BF primitives have two registers per input, and no virtual channels. As a result, a MoT-h-BF network has $R = 6N(N/2^h - 1) + (N/2^h)^2 \cdot 2h \cdot 2^h = O(hN^2/2^h)$ registers.

Table 1 compares register counts of MoT-BF and pure MoT networks for small number of terminals, up to $N = 64$. A 64-terminal MoT-1-BF network has approximately 34% less registers compared to pure MoT.

It is also important to observe the asymptotical behavior of register count. Since h varies between 0 and $\log N$, the number of registers for MoT-h-BF network is asymptotically upper bounded by pure MoT network; and asymptotically lower bounded by pure BF network. For example, if $h = \log N/2$, then $R = O(N\sqrt{N} \log N)$.

N	8	16	32	64
MoT	336	1440	5952	24192
MoT-1-BF	0.62	0.64	0.66	0.66
MoT-2-BF	0.33	0.38	0.40	0.41
MoT-3-BF	0.14	0.20	0.23	0.24

Table 1: Register count of some hybrid MoT-BF networks normalized to MoT with same number of terminals.

4.2 Minimum Latency

Minimum latency is the time in clock cycles, for a packet to travel from source to destination through an empty network. Usually it is averaged over all source-destination pairs, however it does not vary between such pairs in any of the considered networks.

Mesh of Trees A packet travels $\log N$ stages in the fan-out tree, and $\log N$ stages in the fan-in tree. Each stage takes one clock cycle [5]. The overall latency is $L = 2 \log N$.

Virtual-Channel Butterfly The butterfly network has $\log N$ stages of switch nodes. A packet takes three cycles to pass through a regular virtual-channel switch, or at least two cycles to pass through a speculative virtual-channel switch [15]. Assuming regular switches, the minimum latency of a

virtual-channel butterfly is $L = 3 \log N$.

Replicated Butterfly In a replicated butterfly network with r copies, the packets travel through a $\log r$ stage trees before and after the butterfly. Assuming single-cycle switches, the minimum latency is $L = 2 \log r + \log N$.

Hybrid MoT-BF In a MoT-h-BF network, packets pass through $\log N - h$ levels of fan-out and fan-in trees before and after h level butterfly. With single-cycle switches, the minimum latency is $L = 2 \log N - h$.

As h is limited between 0 and $\log N$, the minimum latency of MoT-h-BF network varies between $\log N$ and $2 \log N$ for different hybridization levels.

4.3 Throughput-Area Trade-off

We evaluated maximum throughput of each network by cycle-accurate simulations. For virtual-channel butterfly network, we used the simulator of [7]. For other networks, we use the simulator of [5].

In order to evaluate the maximum throughput provided by each network model, we assume the maximum packet generation rate of one flit per cycle at each input port of the network². At this generation rate, the network will saturate with packets, and the injection and delivery rates will come to balance at the maximum throughput. We assume uniform traffic pattern, which is expected for the memory architecture described in Section 2.1. Uniform traffic pattern is a reasonable assumption due to the use of hashing mechanism, which has an effect similar to randomization that distributes the memory accesses evenly among modules [1, 3, 10].

We simulated networks for $N = 8, 16, 32$ and 64 (see Figure 5 for $N = 8$ and $N = 64$). For each network size, we tuned the throughput by modifying the amount of registers, which are directly related to area cost. Specifically, we modified number of virtual channels v in virtual-channel butterfly, and number of copies r in replicated butterfly networks. As expected, we see that the maximum throughput increases for each network as the number of registers increases, Hybrid MoT-BF network outperforms both BF networks.

4.4 Latency and Throughput vs. Traffic

As network traffic increases, packets will experience longer latencies, and network throughput will saturate. We use a Bernoulli model to generate packets [7], with generation rates varying from 0.1 to 1.0 flits per cycle per port. The net-

²Note that in several other studies of interconnection networks, long data packets may be divided into shorter units, called *flits*. In this network, each packet consists of a single flit.

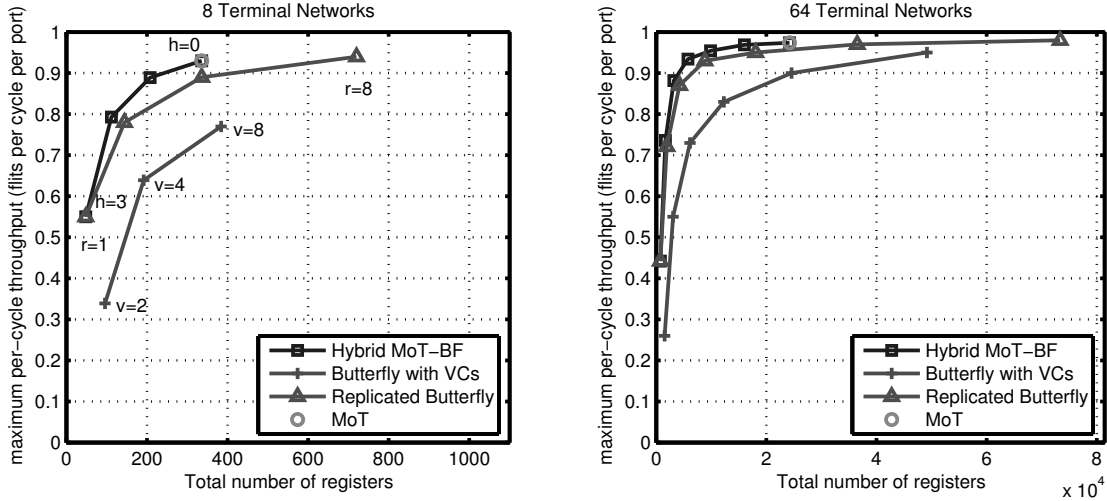


Figure 5: Cost-performance comparison of different network configurations. In each plot, upper left region represents high performance and low area. On the curves, number of virtual channels for VCBF doubles from left ($v = 2$) to right ($v = N$). For RBF, the number of copies doubles from left ($r = 1$) to right ($r = N$). For MoT- h -BF, the hybridization level decreases from left ($h = \log N$) to right ($h = 0$).

work is warmed up until throughput saturates, then marked packets are injected for latency measurement. We are particularly interested in the case when traffic rate, or the on-chip parallelism is high.

Results are shown in Figure 6. At lower traffic rates networks with high hybridization levels have lower latency, because they have fewer stages. At higher traffic rates, packets start to interfere with each other. Networks with lower hybridization levels perform better, and their throughput saturates at higher traffic rates.

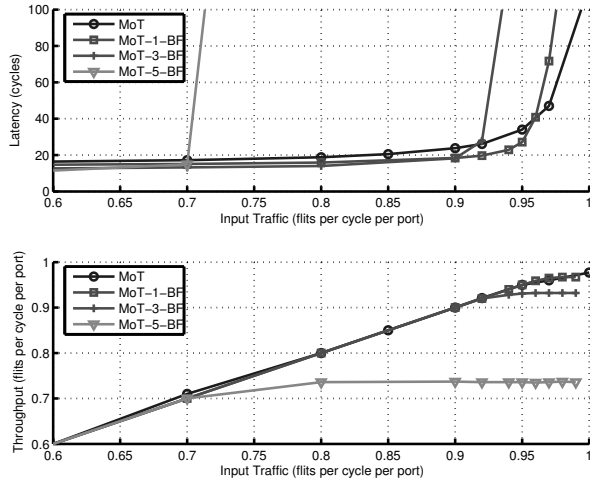


Figure 6: Latency and throughput of 64-terminal hybrid networks as input traffic changes. There is no notable difference among networks when traffic rate is lower than 0.6 flits per cycle. Pure MoT results are also plotted for comparison.

4.5 Post-Layout Throughput

In general, throughput is measured in terms of *Gigabits per second* (Gbps). This value is determined by number of

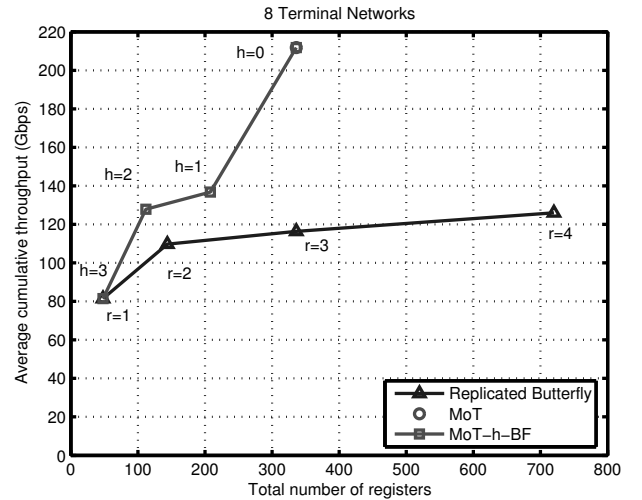


Figure 7: Post-layout throughput of MoT, replicated BF and MoT- h -BF networks. Each flit is assumed to be 32-bits wide.

bits in a flit, number of flits delivered per cycle (*per-cycle throughput*), and clock rate. The number of bits in a flit usually depends on the width of the data path, and it depends on the environment, i.e. the parts of the system that remain outside the network. We assume that this parameter will remain constant with different networks and configurations. Per-cycle throughput depends on network type and architecture. It is usually measured through network simulations (Sections 4.3 and 4.4). The clock rate depends on technology-specific parameters, network and router architecture, and physical layout of the network. Many earlier studies of interconnection networks omit this component, or make safe assumptions because VLSI issues such as wire delays could be neglected at older technologies. On the other hand, some recent studies recognize the importance of the issues and discuss clock rate as well [2].

We create layouts for 8-terminal MoT, RBF and MoT-h-BF networks in order to compute their highest clock frequency and layout-accurate throughput in terms of *Gbps*. We use ARM regular- V_t standard cell library and IBM 90nm (9SF) CMOS technology. Results are shown in Figure 7.

Clock rate of a pure MoT network is the highest among all measured networks, because MoT primitives have shortest delays. Therefore, it has highest throughput in Figure 7. BF primitives perform more complex operation, and this increases their delay. Hybrid networks contain both faster MoT primitives and slower BF primitives. Place and route tools are capable of balancing the wire delays among these primitives to optimize clock rate. As a result, the operation frequency of hybrid MoT-BF networks lies between pure MoT and pure BF networks.

5. CONCLUSION

A hybrid network architecture incorporating mesh-of-trees (MoT) and butterfly (BF) networks is presented. MoT provides superior performance with $O(N^2)$ area cost, where N is the number of network terminals. BF network provides poor performance with $O(N \log N)$ area cost. We replaced inner levels of MoT network with mini-BF networks to build the MoT-BF hybrid network. Based on our analysis, area cost of MoT-BF network lies between pure MoT and BF networks. Under uniform traffic assumption, traffic through mini-BF networks is diluted by preceding fan-out trees. This reduces congestion and related performance loss in mini-BF networks. Simulation results show that MoT-BF performance is between MoT and BF networks up to 64 network terminals. Note that according to [19], each network terminal can support up to 16 light-weight processor.

Operating at same clock rate, a 64-terminal MoT-1-BF network gains 34% area by sacrificing only 0.5% throughput with respect to pure MoT network. It also has approximately 2.5% higher throughput with respect to a RBF network with similar area. Earlier layout-accurate studies observed that the power consumption of MoT network grows at the same rate as the number of cells in the layout [4]. A reduction in register area by hybridization is expected to reduce power consumption accordingly. Hence, MoT-BF hybrid networks support similar UMA-type memory architectures more cost-effectively than pure MoT networks.

Combining simple MoT primitives with more complex BF primitives allows place-and-route tools to balance wire delays accordingly. Resulting layouts on hybrid networks have maximum clock frequencies between pure MoT and pure BF networks. Post-layout throughput of 8-terminal MoT-1-BF is 22% higher than a RBF network with comparable area cost. Pure MoT network has much higher throughput, mostly because of its higher clock rate.

We plan to study MoT-Ring hybrid networks by replacing inner levels of MoT networks with mini-ring networks. Ring network is less expensive compared to BF, but with weaker performance. With proper tuning, ring traffic will be sufficiently diluted, so that congestion and related performance loss in mini-ring networks are reduced. A challenge in this direction is preventing deadlocks. We are studying alternative deadlock-free structures instead of simple rings.

6. REFERENCES

- [1] R. Alverson, et al. The Tera Computer System. In *Proc. Int. Conf. On Supercomputing*, pages 1–6, 1990.

- [2] F. Angiolini, et al. A Layout-Aware Analysis of Networks-on-Chip and Traditional Interconnects for MPSoCs. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 26(3):421–434, 2007.
- [3] P. Bach, et al. Building the 4 processor SB-PRAM prototype. In *Proceedings of the Thirtieth Hawaii International Conference on System Sciences*, volume 5, pages 14–23, Jan. 1997.
- [4] A. O. Balkan, M. N. Horak, G. Qu, and U. Vishkin. Layout-Accurate Design and Implementation of a High-Throughput Interconnection Network for Single-Chip Parallel Processing. In *Proc. IEEE Symp. on High Performance Interconnection Networks*, 2007.
- [5] A. O. Balkan, G. Qu, and U. Vishkin. A Mesh-of-Trees Interconnection Network for Single-Chip Parallel Processing. In *Proc. of the App.-Specific Systems, Architectures and Processors (ASAP)*, pages 73 – 80, 2006.
- [6] W. J. Dally. Virtual-Channel Flow Control. *IEEE Trans. Parallel Distrib. Syst.*, 3(2):194–205, Mar. 1992.
- [7] W. J. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, San Francisco, CA, 2004.
- [8] A. DeHon and R. Rubin. Design of FPGA interconnect for multilevel metallization. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 12(10):1038–1050, Oct. 2004.
- [9] G. R. Goke and G. J. Lipovski. Banyan Networks for Partitioning Multiprocessor Systems. In *Proc. 1st Annu. Symp. Computer Architecture*, pages 21–28, 1973.
- [10] A. Gottlieb, et al. The NYU Ultracomputer—Designing an MIMD Shared Memory Parallel Computer. *IEEE Trans. Comput.*, pages 175–189, Feb. 1983.
- [11] C. P. Kruskal and M. Snir. The Performance of Multistage Interconnection Networks for Multiprocessors. *Computers, IEEE Transactions on*, 32(12):1091–1098, December 1983.
- [12] F. T. Leighton. New Lower Bound Techniques for VLSI. In *Proc. Of the 22nd IEEE Symposium on Foundations of Computer Science*, pages 1–12, 1981.
- [13] F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. Morgan Kaufmann, 1992.
- [14] D. Naishlos, J. Nuzman, C.-W. Tseng, and U. Vishkin. Towards a First Vertical Prototyping of an Extremely Fine-Grained Parallel Programming Approach. *Theory of Computer Systems*, 2003.
- [15] L.-S. Peh and W. J. Dally. A Delay Model and Speculative Architecture for Pipelined Routers. In *Proceedings of the International Symposium on High-Performance Computer Architecture*, pages 255–266, 2001.
- [16] F. P. Preparata and J. Vuillemin. The cube-connected cycles: a versatile network for parallel computation. *Commun. ACM*, 24(5):300–309, 1981.
- [17] J. T. Schwartz. The Burroughs FMP Machine. Ultracomputer Note 5, Courant Institute, NYU, New York, NY, 1980.
- [18] E. Upfal. An $O(\log N)$ Deterministic Packet-Routing Scheme. *Journal of the Association for Computing Machinery*, 39(1):55–70, January 1992.
- [19] X. Wen and U. Vishkin. FPGA-Based Prototype of a PRAM-on-Chip Processor. In *Proc. ACM Computing Frontiers*, Ischia, Italy, May 2008.
- [20] T. T. Ye and G. D. Micheli. Physical Planning for On-Chip Multiprocessor Networks and Switch Fabrics. In *Proc. of the App.-Specific Systems, Architectures and Processors (ASAP)*, pages 97 – 107, 2003.