BENoC: A Bus-Enhanced Network on-Chip for a Power Efficient CMP

Isask'har Walter[1], Israel Cidon[2], and Avinoam Kolodny[2]
Electrical Engineering Department, Technion – Israel Institute of Technology, Israel
[1]zigi@tx.technion.ac.il [2]*{cidon,kolodny}@ee.technion.ac.il*

Abstract—Network-on-Chips (NoCs) outperform buses in terms of scalability, parallelism and system modularity and therefore are considered as the main interconnect infrastructure in future chip multi-processor (CMP). However, while NoCs are very efficient for delivering high throughput point-to-point data from sources to destinations, their multi-hop operation is too slow for latency sensitive signals. In addition, current NoCS are inefficient for broadcast operations and centralized control of CMP resources. Consequently, state-of-the-art NoCs may not facilitate the needs of future CMP systems.
In this paper, the benefit of adding a low latency, customized shared bus as an internal part of the NoC architecture is explored. BENoC (Bus-Enhanced Network on-Chip) possesses two main advantages: First, the bus is inherently capable of performing broadcast transmission in an efficient manner. Second, the bus has lower and more predictable propagation latency. In order to demonstrate the potential benefit of the proposed architecture, an analytical comparison of the power saving in BENoC versus a standard NoC providing similar services is presented. Then, simulation is used to evaluate BENoC in a dynamic non-uniform cache access (DNUCA) multiprocessor system.

Index Terms— Interconnection architectures, On-chip interconnection networks, Network on-Chip support for CMP

1 Introduction

Novel VLSI literature advocates the use of multi-stage Network-on-Chip (NoC) as the main on-chip communication infrastructure (e.g., [2], [4], [7]). NoCs are conceived to be more cost effective than buses in terms of traffic scalability, area, power and performance in large scale systems [3]. Thus, NoCs are considered to be the practical choice for future CMP (Chip Multi-Processor) and SoC (System on Chip) system communication.

During the execution of a typical CMP application, the majority of the traffic delivered by the interconnect involves point-to-point communication. These packets contain cache lines that are being read (or written) by processor cores. However, other kinds of communication should also be facilitated by the CMP interconnect. Examples include L2 cache read requests, invalidation commands for cache coherency, interrupt signals and cache line search operations in DNUCA (Dynamic Non-Uniform Cache Architecture) systems. Although the volume of traffic caused by these operations is relatively small with respect to that of the data read and written by processor cores, the manner in which the interconnect supports them heavily affects both the performance of the system and the dissipated power. While interconnect architectures which solely rely on a network are cost effective in delivering large blocks of data, they have significant drawbacks when other services are required. Primarily, multi-hop networks impose inherent multi-cycle packet propagation latency on the time-sensitive communication between modules (single-hop networks are impractical due to the complexity of the switch required). Moreover, advanced communication services like broadcast (sending information to all modules) or multicast in a network suffer from prolonged latency and involve additional hardware mechanisms or massive duplication of unicast messages. Due to the lack of a central coordination mechanism, the distributed nature of a network is often an obstacle when global knowledge or operation is beneficial.

While current NoC implementations are strictly distributed (heavily borrowing concepts from traditional large scale networks), we argue that the on-chip environment provides the architect with a new and unique opportunity to use "the best of breed" from both on-chip and off-chip worlds. In particular, communication schemes that are not feasible in large scale networks become practical, since on-chip modules are placed in close proximity to each other. Consequently, we propose a new architecture called BENoC (Bus-Enhanced Network on-Chip), which is composed of two tightly-integrated parts: a low latency, low bandwidth specialized bus, optimized for system-wide distribution of control signals, and a high performance distributed network that handles high-throughput data communication between pairs of modules (e.g., XPipes [2], QNoC [4], AEthereal [7]). As the bus is inherently a single hop, broadcast medium, BENoC is proven to be more cost-effective than pure network-based interconnect. Fig. 1 demonstrates BENoC for a cache-in-the-middle CMP. In this example, a grid-shaped NoC serves point-to-point transactions, while global, time critical control messages are sent using the low-cost bus.

BENoC's bus is a synergetic component operating in parallel with the network, improving existing functionality and offering new services. In previous NoC proposals that include a bus (e.g., [12], [13]) the bus typically serves as a layer in the interconnect fabric hierarchy. In [13], each cluster of modules shares a local bus and inter-cluster traffic uses the network. This way, the delivery of data packets to close destinations does not involve the multi-hop network. In [12], the authors suggest a bus-NoC hybrid for a uniprocessor system. By replacing groups of adjacent links and routers by fast bus segments, hop-

Manuscript submitted: 02-May-2008. Manuscript accepted: 04-Jun-2008.
Final manuscript received: 16-Jun-2008.

Posted to the IEEE & CSDL on 7/1/2008
DOI 10.1109/L-CA.2008.11 1556-6056/08/$25.00 © 2008 Published by the IEEE Computer Society

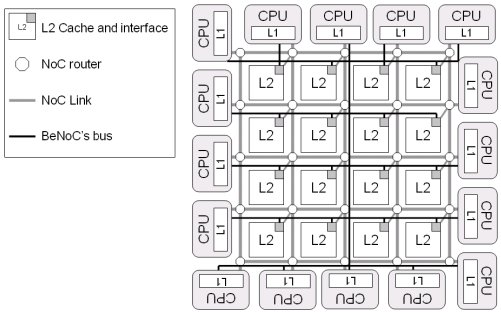

Fig. 1. An example of a BENoC-based CMP System, composed of 16 processors and 4x4 L2 caches.

count is reduced and performance is improved. However, the data which is delivered over the network is later delivered over the bus. In contrast, BENoC's bus is used only to send messages different from those delivered by the network, such as control and multicast messages. In [6], the authors take advantage of the potential benefit that lies in providing special CMP services as part of the interconnect. There, a cache coherence service is embedded into the NoC at the cost of increasing the hardware complexity of routers. Although BENoC can be used for this purpose too, another use of this hybrid architecture is discussed in this paper – searching for migrating cache lines in CMP DNUCA. However, BENoC can provide additional communication services in a cost-effective manner compared to a traditional NoC. For example, considering a L2 read miss event in tiled-CMP system where each core has a private L2 cache, BENoC enables fast, one-hop access to a directory describing the current location of the line being searched. Other services include NoC subsystem control, multicast, anycast and convergecast services and efficient distribution of meta-data. These services are out of the scope of this paper.

2 BENoC FOR CMP DNUCA

In this section, the use of BENoC in a CMP system (Fig. 1) is described. Each processor is assumed to own a local, private (L1) cache while all processors share a distributed L2 cache. As wire latency becomes a dominant factor, the L1 miss penalty is heavily affected by the distance between the processor and the L2 cache bank holding the fetched line. This observation gave rise to the DNUCA approach: instead of statically allocating cache lines to L2 locations, cache lines are moved towards processors that access them [9], [10].

A major difficulty in implementing DNUCA is the need to lookup cache lines: whenever a processor needs to conduct a line fill transaction (fetch a line into its L1 cache), it needs to determine its location, i.e., the identity of the L2 cache bank/processor storing its most updated copy. In order to keep track of the current sharers/owner of a line, each L2 cache line may include some extra bits [1], forming a distributed directory. As statically-located entries would ruin the benefits of local access in DNUCA, these bits migrate together with the line. In a network-based interconnect, the line can be sought using multiple unicast messages [5]. Alternatively, the interconnect designer may add some extra logic within the network infrastructure to facilitate an in-band deadlock-free broadcast service (e.g., [8]). BENoC architecture offers a more efficient alternative, by using the bus to broadcast the query to all cache banks in a fast, one-hop transmission. The particular cache storing the line acknowledges the request on the bus, and simultaneously sends the line's content over the high-bandwidth NoC. As queries are composed of short meta-data (e.g., the initiating processor's ID and the line's address), they do not create a substantial load the bus.

The proposed scheme has two main advantages: First, it reduces the power consumption of the interconnect as a single bus transaction performs the broadcast operation, instead of multiple unicast messages in the NoC. Second, the system performance is improved as the time-critical line search is performed over a dedicated single-hop medium instead of over a multi-hop network. These aspects are explored in the following two sections.

3 ANALYSIS

In this section, the energy required for a broadcast operation in a traditional NoC and in BENoC is approximated. For simplicity, the NoC is assumed to have a mesh topology. Analysis of other multi-hop NoCs is conducted in a similar manner. The following notation is used:

n = The number of modules in the system
ΔV = Voltage swing [V]
C_0 = Global wire capacitance per unit of length [F/mm]
P = Tile size [mm]
C_{ld} = NoC link driver input capacitance [F]
C_{link} = NoC link capacitance [F]
C_{bd} = Bus driver input capacitance [F]

The time needed for a driver to charge a capacitor is modeled using the following equation [14]:

$$T = \frac{\tau}{C_{in}} C_{Load} + \tau \qquad (1)$$

where C_{in} is the driving buffer's input capacitance and C_{load} is the load's capacitance. The constant τ, which is determined by the technology, is the product of the effective resistance and the input capacitance of an inverter: $\tau \simeq R_{inv} C_{inv}$. The energy required to charge C_{load} is

$$E = \Delta V^2 \cdot C_{load} \quad . \qquad (2)$$

First, the latency and energy of a broadcast transaction in a NoC-based system which relies on multiple unicast messages is approximated. Assuming a NoC link is P millimeters long, its capacitance is $C_{link} = P \cdot C_0$. Using (1) and the above definitions, the time required for a link driver to transmit a single bit is $T_{link} = (\tau / C_{ld})(C_{link} + C_{in}) + \tau$, where C_{in} is the input capacitance of the input port to which the link is connected. Since a broadcast message has to travel at least $\sqrt{n}$ modules away from the source, the minimal time to complete the broadcast (neglecting delay within routers) is

$$T_{net} = \sqrt{n} \cdot T_{link} = \sqrt{n}\left(\frac{\tau (P \cdot C_0 + C_{in})}{C_{ld}} + \tau \right). \qquad (3)$$

Note that (3) underestimates the broadcast latency, as messages are withheld at least one clock cycle in each

router along their path. If no priority is given to such packets, they might also be delayed due to congestion.

In order to calculate the total energy needed for NoC broadcast, the number of packet transmissions is determined. In a regular mesh, a source node may have at most 8 modules at a distance of one, 16 modules two hops away, 24 modules three hops away and so on. In the energy-wise best case, the broadcasting module is located exactly in the middle of the mesh. It therefore has to send 8 messages that would each travel a single link each, 16 messages that travel two links, and in general, 8j messages to a distance of j hops, until transmitting a total of n-1 messages. It can be easily shown that if $\sqrt{n}$ is an integral, odd number, then the Manhattan distance between the module in the middle of the mesh and the ones in its perimeter is exactly $D_{\max}=\left(\sqrt{n}-1\right)/2$. Since a message transmitted to a destination j hops away has to traverse j router-to-router links, the minimal number of transmissions required to complete the broadcast is

$$K=8\cdot 1+16\cdot 2+24\cdot 3+\ldots+8D_{\max}\cdot D_{\max}=8\sum_{j=0}^{D_{\max}} j^2 \qquad (4)$$

Consequently, the lower bound of the total energy consumed by a single broadcast operation according to (2) is

$$E_{net}=\Delta V^2\cdot K(C_{ld}+C_{link}+C_{in})\cdot \qquad (5)$$

Similarly, the latency and energy that characterize a broadcast on a bus is now evaluated. The bus is assumed to be composed of $\sqrt{n}$ horizontal sections (of length $\sqrt{n}\cdot P$ each), connected together using a vertical segment of the same length (Fig. 1). As the total bus length is approximately $\left(\sqrt{n}+n\right)P$ long, and assuming that it is connected to n loads of C_{in} each, its total capacitance is approximately $C_{bus}\simeq\left(\sqrt{n}+n\right)PC_0+nC_{in}$. The resulting broadcast transmission delay according to (1) is

$$T_{bus}=\frac{\tau}{C_{bd}}C_{bus}+\tau=\frac{\tau}{C_{bd}}\left(\left(\sqrt{n}+n\right)PC_0+nC_{in}\right)+\tau\ . \qquad (6)$$

Using (2), the total energy required to drive the bus is

$$E_{bus}=\Delta V^2(C_{bus}+C_{bd})=\Delta V^2\left(\left(\sqrt{n}+n\right)PC_0+nC_{in}+C_{bd}\right). \qquad (7)$$

Clearly, the bus driver has to be much more powerful (and thus, energy consuming) than a link driver. In order to choose an appropriate sizing for the bus driver, the parameter β, which reflects the network-to-bus broadcast latency ratio, is defined: $T_{net}/T_{bus}=\beta$. Using (3) and (6) the bus driver size for achieving a desired latency ratio β is determined:

$$C_{bd}=\frac{\tau\left(\sqrt{n}PC_0+nPC_0+nC_{in}\right)}{\frac{\sqrt{n}}{\beta}\left(\frac{\tau(P\cdot C_0+C_{in})}{C_{ld}}+\tau\right)-\tau} \qquad (8)$$

Finally, the total energy consumption required for a bus broadcast is established using (7) and (8):

$$E_{bus}=\Delta V^2\left(\left(\left(\sqrt{n}+n\right)PC_0+nC_{in}\right)+\frac{\tau\left(\sqrt{n}PC_0+nPC_0+nC_{in}\right)}{\frac{\sqrt{n}}{\beta}\left(\frac{\tau(P\cdot C_0+C_{in})}{C_{ld}}+\tau\right)-\tau}\right).$$

For simplicity, we assume a single, centralized arbiter which is connected to the system modules using request-grant lines. Due to the point-to-point nature of these wires, the energy consumption and the signal propagation delay of the arbitration mechanism are negligible with respect to those of the chip-wide high capacitance bus. The implementation and analysis of more sophisticated arbiters is left for future work. In order to complete the analysis, typical values for the various electrical parameters for 0.65um technology [15] are used (C_0=205.94e-15F; R_{inv}=8760Ω; $C_{inv}=C_{in}$=7.36e-16F; C_{ld}=3.68e-14F; ΔV=1V). The tile size (P) is set to 1mm.

Fig. 2 shows the energy required for unicast and broadcast transmissions in a NoC. It also shows the energy required for a transmission in BENoC for two bus speeds (values of β). As expected, the bus is no match for the NoC when a message is delivered to a single destination. However, when broadcast operations are compared, the bus is considerably more energy efficient than the network, as shown by the "network broadcast" curve compared with the "bus transaction" curves, for system size n of ~25 or more.

4 Experimental Results

In this section, we evaluate the performance of a large scale DNUCA CMP, comparing a classic NoC with a BENoC infrastructure. Two time-critical operations are addressed. The first is a basic line-fill ("read") transaction, performed by a processor that reads a line into its L1 cache. If an L2 cache has a valid copy of the line, it must provide its content to the reading processor. If the most updated copy resides in a L1 cache of another processor, it is asked to "writeback" the line. Else, the line is fetched from a lower memory hierarchy level (L3 cache or memory). The second transaction being addressed is the read-for-ownership ("read-exclusive") transaction. While similar to the basic line-fill operation, it also implies that the reading processor wishes to own the single valid copy of that line for updating its content. In order to complete the transaction, all other L1 copies of the line (held by an owning processor or by sharers) must be invalidated.

In a classic DNUCA implementation, the processor has to lookup the line prior to the read/read exclusive operation. When a regular NoC is used, the line is sought using multiple unicast messages, while in BENoC the search is conducted over the bus. In this work, a distributed directory model is assumed: each L2 cache line includes some

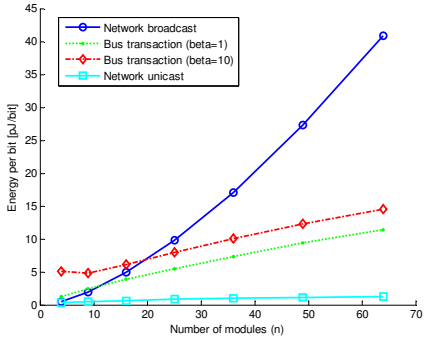

Fig. 2. Energy consumption in NoC and BENoC. The energy consumed by a network unicast and broadcast, and by a bus transmission.

extra (directory) bits to keep track of the current sharers/owner of the line [1]. As explained in Section 2, a static directory would render the DNUCA policy useless, hence these bits migrate together with the line. The simulated system consists of 16 processors and 64 L2 cache tiles (80 modules in total). The modules are arranged as depicted in Fig 1, while the 4x4 cache array is replaced by an 8x8 array of banks. The network link capacity is set to 50Mflits/s. BENoC's bus is assumed to be made of a single segment, 32 bits wide. A round-robin arbiter is placed around the center of the chip.

In order to evaluate the proposed technique, two simulators are used. The BENoC architecture is simulated using OPNET [16]. The model accounts for all network layer components, including wormhole flow control, virtual channels, routing, buffers and link capacities. In addition, it simulates the bus arbitration and propagation latencies. The DNUCA system is modeled using the Simics [11] simulator and SPLASH-2 benchmarks.

Fig. 3 presents the improvement achieved by the BENoC architecture compared to a classic NoC, in terms of linefill transaction time, application speedup and interconnect energy. BENoC considerably reduces the average transaction time (Fig. 3a) even when a relatively slow (thus, power efficient) bus is used (β=0.5). This is because the analysis presented above underestimates the network latency. In a NoC-based system, broadcast messages are likely to collide, as they compete for shared network resources while in BENoC, a single arbitration phase is required prior to the bus access. This effect becomes more dominant as the number of broadcasting cores is increased. In addition, NoC routers introduce latency even when no congestion occurs. The decreased linefill latency reduces the total application execution time by 32% on average (Fig. 3b).

Fig. 3c depicts the interconnect energy saving achieved in BENoC. By using the bus for searching cache lines, the total interconnect energy consumption is reduced by 16% on average. In addition, the energy consumed by the processors during execution of the application is reduced due to the shortened execution time. For example, if the power of a processor while waiting for data from the cache is 50% of its full active power, then a 32% speedup of the application would decrease the processor energy consumption by 16%.

5 SUMMARY

A salient feature of on-chip systems is the proximity of all components within a distance of several millimeters, which enables low-latency communication among them. While macro networks generally cannot benefit from out-of-band communication and use their standard links for all operations, NoCs can leverage a side-bus to improve system functionality.

The bus-enhanced NoC architecture optimizes the communication infrastructure by combining a customized bus and a NoC. The bus circumvents principal weaknesses of the NoC, such as latency of critical signals, complexity and cost of broadcast transactions, and operations requiring global knowledge or central control. While the

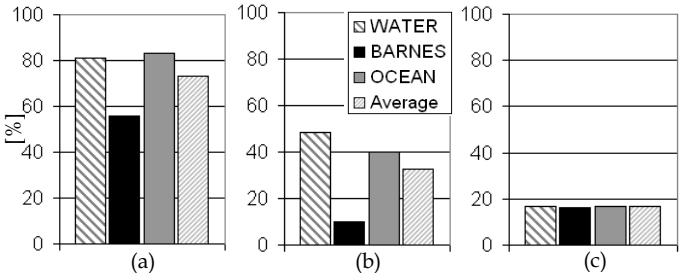

Fig. 3. Performance improvement and energy reduction in BENoC compared to a NoC-based CMP: (a) Linefill transaction time reduction; (b) Execution time speedup; and (c) Interconnect energy saving (β=0.5).

BENoC architecture provides many opportunities, in this paper it is shown to be superior to a classical NoC in terms of performance and power for a CMP DNUCA system. Analysis shows that BENoC's advantage over NoC starts at relatively small system size and becomes very significant as system size grows. Simulation reveals that a relatively slow, power efficient bus improves the system's performance and reduces the interconnect energy.

REFERENCES

[1] B. M. Beckmann and D. A. Wood, "Managing wire delay in large chip multiprocessor caches", MICRO 37, pp. 319-330, Dec. 2004

[2] D. Bertozzi and L. Benini, "Xpipes: A Network-on-Chip Architecture for Gigascale Systems-on-Chip", Circuits and Systems Magazine, IEEE Volume 4, Issue 2, pp. 18-31, 2004

[3] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "Cost Considerations in Network on Chip", Integration - the VLSI Journal, Volume 38, pp. 19-42, 2004

[4] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "QNoC: QoS Architecture and Design Process for Network on Chip", Journal of Systems Architecture, Volume 50, pp. 105-128, February 2004

[5] E. Bolotin, Z. Guz, I. Cidon, R. Ginosar, and A. Kolodny, "The Power of Priority: NoC based Distributed Cache Coherency", Proc. First Int. Symposium on Networks-on-Chip (NOCS), pp. 117-126, 2007

[6] N. Eisley, L.S. Peh, and L. Shang, "In-Network Cache Coherence", Proc. 39th Int. Symposium on Microarchitecture, pp. 321-332, 2006

[7] K. Goossens, J. Dielissen, and A. Radulescu, "AEthereal Network on Chip: Concepts, Architectures, and Implementations", IEEE Design and Test of Computers, pp. 414-421, 2005

[8] Y. Jin, E. J. Kim, and K. H. Yum, "A Domain-Specific On-Chip Network Design for Large Scale Cache Systems", Proc. 13th Int. Symp. on High-Performance Computer Architecture, pp. 318-327, 2007

[9] C. Kim, D. Burger, and S.W. Keckler, "An Adaptive, Non-Uniform Cache Structure for Wire-Delay Dominated On-Chip Caches", 10th International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 211-222, October, 2002

[10] C. Kim, D. Burger, and S.W. Keckler, "Nonuniform Cache Architectures for Wire Delay Dominated on-Chip Caches", IEEE Micro, 23:6, pp. 99-107, November/December, 2003

[11] P.S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, and G. Hallberg "Simics: A full system simulation platform", IEEE Computer, 35(2):50–58, Feb. 2002

[12] N. Muralimanohar and R. Balasubramonian, "Interconnect design considerations for large NUCA caches", Proc. 34th annual International Symposium on Computer architecture, pp. 369-380, 2007

[13] T.D. Richardson, C. Nicopoulos, D. Park, V. Narayanan, Y. Xie, C. Das, V. Degalahal, "A Hybrid SoC Interconnect with Dynamic TDMA-Based Transaction-Less Buses and On-Chip Networks", Proc. 19th International Conference on VLSI Design, pp. 657-664, 2006

[14] I. Sutherland, R.F. Sproull, and D. Harris, "Logical Effort: Designing Fast CMOS Circuits", The Morgan Kaufmann Series in Computer Architecture and Design, ISBN: 978-1-55860-557-2

[15] Predictive Technology Model, http://www.eas.asu.edu/~ptm

[16] OPNET Modeler, www.opnet.com