A Hybrid Packet/Circuit-switched Router to Accelerate Memory Access in NoC-based Chip Multiprocessors

Abbas Mazloumi[1], Mehdi Modarressi[1,2]
[1]School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran, Iran
[2]School of Computer Science, Institute for Researches in Fundamental Sciences, Tehran, Iran
{a.mazloumi, modarressi}@ut.ac.ir

Abstract— **Modern chip multiprocessors will feature a large shared last-level cache (LLC) that is decomposed into smaller slices and physically distributed throughout the chip area. These architectures rely on a network-on-chip (NoC) to handle remote cache access and hence, NoCs play a critical role in optimizing memory access latency and power consumption. Circuit-switching is the most power- and performance-efficient switching mechanism in NoCs, but is not advantageous when the packet transmission time is not long enough compared to the circuit setup time. In this paper, we propose a zero-latency circuit setup scheme to make circuit-switching applicable in transferring individual data packets. The design leverages the fact that in CMPs with distributed LLC (where a considerable portion of the on-chip traffic is composed of remote LLC access requests and data responses), every response packet is sent in reply to a request packet and traverses the same path as its corresponding request, but at the backward direction. The short request packets, then, are responsible to reserve a path for their corresponding response packets. This NoC tries to reduce conflict among circuit paths by considering conflicts in backward direction during request packet routing, backed by a run-time technique to resolve conflicts when circuits are actually set up. Experimental results show that the proposed NoC architecture considerably reduces average packet latency that directly translates to faster memory access.**

Keywords—Network-on-chip; Circuit-switching; Chip-multiprocessor; Memory-access;

I. Introduction

The exponential increase in transistor count, coupled with the ever increasing demand for higher performance in embedded, desktop, and server computers, have moved the semiconductor industry toward many-core chip multiprocessors (CMPs) and systems-on-chip (SoCs). Major semiconductor manufacturers already fabricate chips with few tens to hundreds of cores and according to the ITRS projections, chips with several hundreds to thousands of cores are likely to appear in near future [1].

A modern chip multiprocessor (CMP) features tens of processing cores, each with one or more levels of private caches, backed by a large shared last-level cache (LLC). The most effective solution to mitigate the long LLC access latency in large CMPs is to divide the LLC into smaller slices and physically distribute them throughout the chip along with the cores [2][3]. The network-on-chip (NoC) in such systems provides a communication infrastructure to access remote LLC slices and maintaining private caches coherent.

Since many modern workloads spend a considerable portion of their execution time on on-chip cache accesses [3], optimizing the NoC for cache access is critical to achieve high performance. Reducing the NoC power consumption is also very important for scaling up the number of nodes in future many-core systems.

The on-chip traffic of a typical CMP workload is mainly composed of short fetch request messages and long response messages carrying one or more cache blocks. Although methods that take the message type into account for VC allocation and arbitration can be found in the literature [4][5], almost all existing NoCs treat request and response messages quite equally from the prospective of switching mechanism, which is one of the most important features of a NoC.

In this paper, we propose a NoC architecture that adopts packet-switching for short request messages and circuit-switching for longer response messages. In this design, response packets and thus, a large portion of on-chip traffic, take advantage of the superior performance of circuit-switching. Circuit-switching also removes the need for buffering data at each intermediate router and since buffers account for a significant portion of router power consumption, the proposed NoC can also reduce NoC power consumption.

In order to setup circuit for each individual packet, the first necessary step is to eliminate the main performance drawbacks of circuit-switching, i.e. prohibitive long circuit setup time and low resource utilization. To eliminate the former problem, we leverage the fact that every request message is followed by a response message that traverses the path between the same endpoint nodes, but in the reverse direction. The request packet, then, reserves a circuit for its corresponding response message when it moves to the destination node. Response messages, consequently, experience zero circuit setup time.

Resolving the second problem, our NoC is avoided by not reserving the circuit by the request packets, but request packets just find the circuit path and do not specify the exact circuit setup time; the circuit is actually established along the found path by a probe message when the response packet is ready to be sent. In addition to using a conflict-aware routing for request packets that tries to find a path with minimum probability of conflict with other circuits in the backward direction, potential conflicts among circuits are handled by a run-time mechanism when response messages are traversing circuits.

The rest of the paper is organized as follows. We present the router architecture and circuit setup scheme in Section 2.

978-3-9815370-4-8/DATE15/©2015 EDAA

Experimental results are presented in Section 3 and finally, Section 4 concludes the paper.

II. Proposed NoC Architecture

A. Hybrid packet/circuit-switched router structure

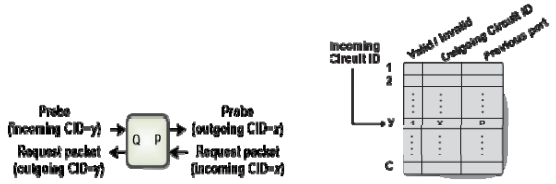
Each router in our design implements the baseline wormhole-switching to handle control packets. The routers are also equipped with some extra logic to support circuit-switching for longer response packets.

The routing unit of the wormhole-switched part of the NoC is modified to take its additional task, which is reserving circuits in the backward direction, into account. To achieve this goal, a fully adaptive routing is used and deadlock freedom is guaranteed by using two virtual channels under the virtual sub-network deadlock avoidance scheme [6].

The routing logic first examines the availability of output ports in the forward direction. The routing in particular should prevent request packets to be blocked behind a long response message by not selecting output ports that are currently allocated to circuits. If all ports have the same level of availability, the one that has less possibility of conflict for the anticipated response message is selected. The selection measure is the number of already reserved circuits on each designated port.

B. Circuit reservation

A maximum of C circuits can be reserved on each router output port and a local circuit identifier (CID) is assigned to each reserved circuit. Figure 1.a shows the procedure of reserving a circuit and reclaiming it at a NoC node. In this figure, a request packet comes from input port P with input CID x (which is assigned by the upstream router) and leaves the router through port Q with a new CID y. When the probe message of the corresponding response packet returns to this router, it comes with y as CID and uses the CID to retrieve the next port (P) and the next CID (x) from the table and continues setting up the path.



(a) (b)

Fig. 1. (a) The message sequence of circuit reservation and setup and (b) the reservation table of output port Q

The path of a reserved circuit is kept by all routers along the path in a distributed manner. A reservation table with C entries is associated with each router output port to keep reserved circuits and is indexed by the local CID of that port. Figure 1.b shows a reservation table and the entry related to the circuit of Figure 1.a.

Once output port Q is selected by the routing logic to forward the request packet, a free CID of that port is assigned to it by selecting one of the free entries of the reservation table of port Q and using its index as the packet CID. The previous CID of the packet and its input port is stored in the entry to keep the reserved circuit path.

If all CIDs are taken by previous request packets, the request cannot advance and should wait until a circuit is set up and its CID become available. If a request waits more than a predefined threshold, it abandons circuit setup and a cancellation probe message goes down the path to release the reserved CIDs. The request and response packets then should use the packet-switched sub-network.

C. Circuit setup

Once a response packet gets ready, a probe message starts traversing down the circuit n cycles ahead of the data packet to reclaim the reserved circuit.

Last-level caches of most processors adopt a serial tag and data lookup to reduce energy consumption [7]. For such LLCs, probe is sent as soon as the LLC tag lookup indicates a hit and leads the response message by the time between the end of the tag and data lookup.

The probe simply consists of the CID of the circuit and is directed over a dedicated light-weight control network down to the requesting node. The control network should be wide enough to carry CIDs plus two control bits (6 bits in our experiments).

On receiving the probe message at some cycle t, the probe accesses the reservation table of the incoming port by using its CID as index to find the output port through which it should continue circuit setup. It then takes required time slots on that port for the data packet by appropriately setting the output port slot table. Based on the latency analysis in [8], the probe can perform these simple tasks in a single NoC cycle.

The connection will be established from cycle t+n to t+n+l, where l is the packet size and n, as mentioned before, is the number of cycles by which probe leads the packet. The probe then continues to the next router through the output port using the new CID taken from the reservation table. It also releases its reservation table entry (CID).

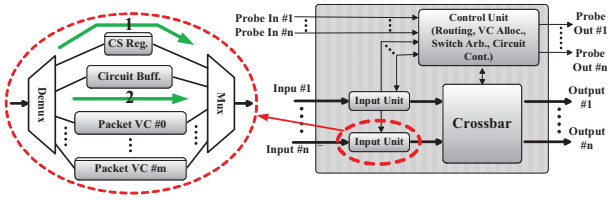


Fig. 2. The router architecture and details of one input port

Figure 2 shows the router architecture with the details of one input port. At each input port, packet-switched data are get buffered at one of the packet-switched VCs based on the VC allocation made by at upstream router and are forwarded to the next node after a sequence of routing, arbitration, VC allocation and flow control operations.

On successful circuit setup in a router, path 1 in Figure 2 is used during the l-cycle interval to store received flits in a register and send them to the next node along the circuit at the next cycle in a pipelined manner. In this case, the need for power-hungry and time consuming buffering, routing, VC allocation and switch arbitration is eliminated. However, if the required time slots on the requested output port is already assigned to another circuit, the router sets up path 2 to locally store flits and forward them immediately when the port becomes available.

Figure 3 shows these scenarios for two circuits: circuit X from A to B and circuit Y from C to D. The circuits have a link in common at the west output port of node E. The probe of circuit X and circuit Y are issued at time t and t+1, respectively. Assuming n=2 and l=5, the first flit of circuit X is injected at cycle t+2 and the first flit of circuit Y is injected at cycle t+3.

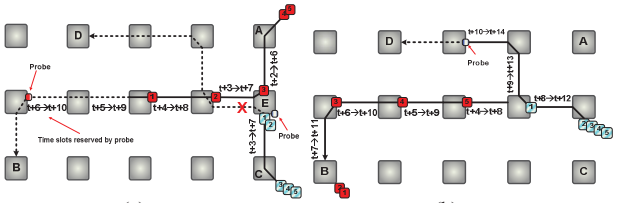


Fig. 3. NoC snapshot at cycles (a) t+4 and (b) t+8 for circuit X between nodes A and B and circuit Y between nodes C and D. Solid line represents established path and dashed line shows the circuit path that should be established by the probe message. Each link is tagged by the time slots allocated to a circuit (if any)

Figure 3.a shows the NoC snapshot at cycle t+4, when circuit Y collides with circuit X at node E. The flits of circuit X follow their probe and traverse down the circuit normally.

The probe message and flits of Y, however, are blocked and get buffered at node E. Since the required output port will be available at cycle t+8, the probe of circuit Y resumes traversal at t+6 (because it has to always lead the data by 2 cycles) to continue setting up the circuit along the path. Figure 3.b shows the snapshot of the NoC at cycle t+8 and clarifies how circuit Y flits continue lagging behind the probe by 2 cycles after the blocking is resolved.

Packet- and circuit-switched data compete for the NoC links, but circuit-switched flits are prioritized, because they don't have flow control and cannot wait for links. If a packet waits for an output port behind a circuit, the router delays the probe of the next circuit of that output port (if any) for a few cycles (in much the same way as when an inter-circuit conflict occurs) to allow packet-switching part to use that output port and prevent packet starvation.

To handle the very rare deadlock condition in circuit-switched sub-network, if the stay of a response message in a CS buffer exceeds a threshold, the response message is switched to the packet-switched part and a probe message is sent to release the CID along the rest of the path.

III. Experimenatl Results

A. Evaluation methodology and environment

The proposed NoC is compared against the state-of-the-art packet-switched router in a mesh topology. The packet-switched router forwards packets in two cycles; one cycle for look-ahead routing (XY routing) and arbitration, followed by one cycle for crossbar and link traversal. To evaluate performance (latency and throughput) and power characteristics, we use Booksim [9], a detailed cycle-accurate NOC simulator. The evaluations are done under a set of standard synthetic traffics, as well as the Netrace library [10] that models the PARSEC traffic.

We consider a 64-core CMP arranged as an 8×8 mesh. In the packet-switched part, each port has two virtual channels per port with 8-flit buffers. The circuit-switched part consists of a register and a buffer to keep the circuit-switched data in case of a conflict. The power results are calculated by Orion-3 power library [11] which gives acceptable accuracy for 2D mesh.

B. Evaluation

We evaluate our method using four standard synthetic traffic profiles: bit-rotate, hot flow [12], transpose, and uniform. These synthetic traffic profiles provide insight into the relative strengths and weaknesses of the considered NoC. The request packet length is set to one flit, while the length of the response packet is set to five flits. Traffic profiles (and the injection rate) determine the injection of request packets into the network. In case of a hit, a response packet gets injected into the network 10 cycles after the arrival of the request packet to the destination. We consider 20% miss ratio for the LLC with 100 cycle penalty. Using CACTI, we estimated the tag and data lookup delays of the LLC to be one and four cycle(s), respectively, thereby a probe is sent to set up the circuit three cycles ahead of the response packet.

The average flit latency for different request injection rates under uniform traffic is shown in Figure 4. As the figure shows, the proposed method reduces the network delay before the saturation point. The gain margin diminishes near and at the saturation point, but we never operate a network-on-chip in such conditions.

We do not present the graphs for the other three traffic patterns due to the limited space and only show the average flit latency under them at a moderate load in Figure 5.a.

Figure 5.b displays the energy consumption of the networks at a selected injection rate (0.05 packet/node/cycle, which is right before the saturation point of the mesh under uniform traffic) and shows that the proposed method considerably reduces the NoC energy consumption. The main source of this reduction is the elimination of buffering, routing, and arbitration for most of the response messages. The results consider the probe message power consumption.

Figure 6 compares the proposed NoC with the packet-switched network under the PARSEC traffic profiles modeled by Netrace. The request packet length is one flit and the response messages are 9-flit long.

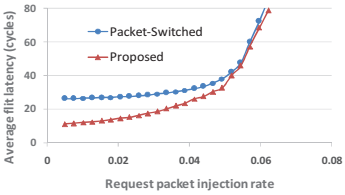
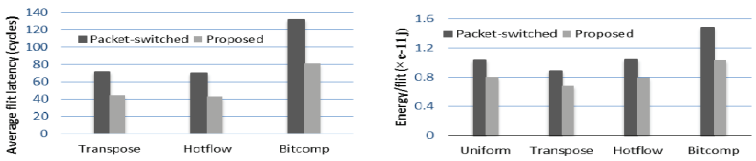
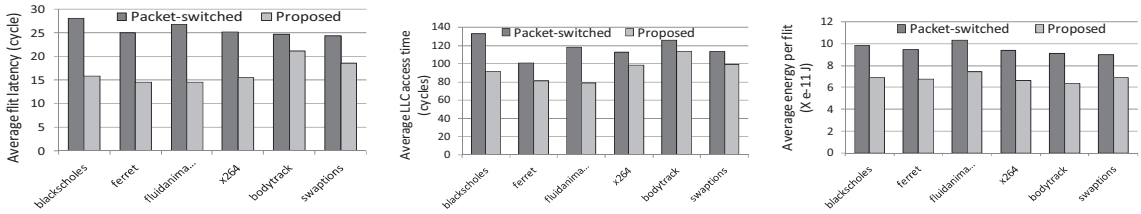


Fig. 4. Average flit latency under uniform traffic



(a) (b)

Fig. 5. (a) Average flit latency and (b) energy per flit under synthetic traffic patterns at the injection rate of 0.04 and 0.05 request/node/cycle respectively



(a) (b) (c)

Fig. 6. (a) Average flit latency, (b) average memory (LLC) access time and (c) energy per flit under Netrace traffic model

As the figure 6.a shows, the proposed NoC yields 34% average reduction in flit latency that translates to 18% reduction in average memory (LLC) access time, as shown in Figure 6.b. The memory access time is calculated as the cumulative latency of the request and response messages and also the queuing time in the source and destination nodes. Following the same trend, the energy consumption across all benchmarks is reduced by 23%, on average, as shown in Figure 6.c.

The proposed NoC requires a probe routing unit with 6-bit narrow, bufferless routers and links, for establishing and maintaining circuits. As most of the area of a router is due to buffers and crossbars, the difference between the area of a conventional packet-switched NoC and the proposed NoC, both having 128-bit links, using the area model proposed in [13] is 4%.

IV. Conclusions

This paper presented a hybrid circuit/packet-switched NoC that accelerates memory access in shared memory CMPs. In this NoC, short control packets are directed to the packet-switched part, whereas longer data packets use circuit-switching and travel over circuits that are reserved by their corresponding request packet. The circuits are not established immediately by the request packet, but the path is stored in routers and a probe message reclaims the circuit a few cycles before the actual data transmission. The routing of request packets not only considers the congestion of the forward direction but also takes into account the availability of free link time at the opposite direction where the circuit will be established for the corresponding response packet later. This approach to circuit setup solves the two main problems of the traditional circuit-switching, i.e the long setup time and low resource utilization. Experimental results showed 33% decrease in average flit latency across a set of synthetic and realistic workloads that translates to 18% reduction in memory access time.

References

[1] N. Hardavellas, et al., "Toward Dark Silicon in Servers," *IEEE Micro*, vol. 31, no. 4, pp. 6-15, 2011.

[2] B. K. Daya, et al., "SCORPIO: a 36-core research chip demonstrating snoopy coherence on a scalable mesh NoC with in-network ordering", in *Proc. of ISCA*, 2014.

[3] N. Hardavellas, et al., "Reactive NUCA: near-optimal block placement and replication in distributed caches", in *Proc. of ISCA*, pp. 184-195, 2009.

[4] A. Naeem, A. Jantsch and Z. Lu, "Architecture Support and Comparison of Three Memory Consistency Models in NoC based Systems", in *Proc. of Euromicro DSD*, 2012.

[5] Bolotin, Evgeny, et al., "QNoC: QoS architecture and design process for network on chip"," *Journal of Systems Architecture*, Vol. 50, No. 2, pp. 105-128, 2004.

[6] W. J. Dally, and B. Towles, *Principles and practices of interconnection networks*, Morgan-Kaufmann Publishers, 2004.

[7] D. Sanchez and C. Kozyrakis. The ZCache: Decoupling Ways and Associativity. In *Proc. of the 43rd Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 187-198, 2010.

[8] T. Krishna, et al., "Breaking the On-Chip Latency Barrier Using SMART", in *Proc. of HPCA*, 2013.

[9] Booksim NoC simulator, *http://nocs.stanford.edu/booksim.html*, 2013.

[10] J. Hestness, S. W. Keckler. "Netrace: Dependency-Tracking Traces for Efficient Network-on-Chip Experimentation." Technical Report TR-10-11, The University of Texas at Austin, May 2011.

[11] A. Kahng, et al., "Explicit Modeling of Control and Data for Improved NoC Router Estimation", in *Proceedings of DAC*, 2012.

[12] M. Modarressi, et al., "Virtual Point-to-Point Connections in NoCs", *IEEE on Computer-Aided Design for Integrated Circuits and Systems*, vol. 29, pp. 855-868, 2010.

[13] M. Modarressi, et al., "Application-Aware Topology Reconfiguration for On-Chip Networks", *IEEE Transactions on Very Largescale Integrated Circuits and Systems,* Vol. 19, No. 11, pp. 2010-2022, Nov. 2011.