Development of Efficient On-Chip Networks for High-Performance Processors

Duraid Madina
RIKEN ASI
Ono-cho 61-1, Tsurumi
Yokohama, Japan 230-0046

1. INTRODUCTION

High-performance computers have historically been one of the main drivers behind high-performance networks. A significant body of work into network toplogies, signalling techniques, routing, broadcast and reduction algorithms exists and continues to grow.

However, to this day, modern high-performance computers, including the RIKEN "Next-Generation" supercomputer (NSC), are partitioned such that the high-performance interconnect is entirely distinct from the processors. In the case of the NSC, its processors contain at their heart a crossbar switch which has no interaction with the external network except through a dedicated I/O port. That is to say, it is not possible for individual processor cores to address each other through the network in a uniform manner; instead, cores in the same silicon processor "address" each other through the on-chip crossbar, while cores in different processor chips must traverse through the local crossbar, then through the interconnect, and finally through the destination chip's crossbar.

Currently, the fact that the interconnect fabrics of modern supercomputers do not reach into the processors themselves is not a major concern, since the number of cores in a processor is on the order of 10 or so.

2. ON-CHIP NETWORKS

In the near future (perhaps 5 to 10 years from today), the number of cores on processors intended for scientific applications is likely to approach 100. In this case, existing approaches to processor core interconnection become difficult. Simply using buses becomes difficult since even modest amounts of traffic (e.g. instruction cache fills) could quickly overwhelm a shared bus. Widening or splitting buses in an ad-hoc manner has its own difficulties, and so another common approach is to interconnect processor cores through a large crossbar switch.

This technique has been adopted by a number of commercial processor designs similar to a hypothetical 8-core processor illustrated in Figure 1. Assuming the crossbar has an area of approximately 20mm^2, it is likely to amount to no more than 5% of the chip area, a reasonable physical cost. Since the any-to-any connectivity of the crossbar completely avoids performance bottlenecks related to traffic patterns. However, since the area, power consumption and latency of a crossbar switch grow as the square of the number of crossbar ports, this technique cannot be used to connect the 16, 64 or hundreds of processor cores likely to appear in future high-performance processors. Moreover, the crossbar

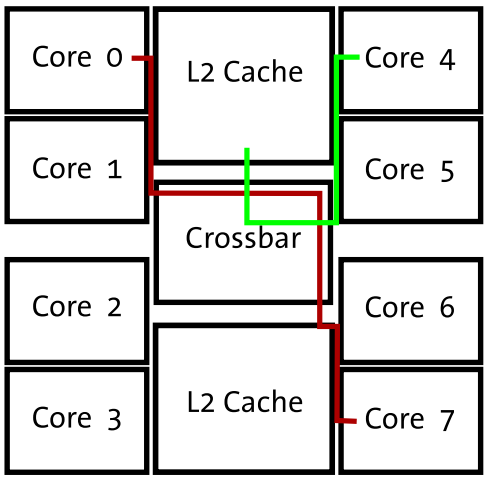

Figure 1: A hypothetical 8-core processor with split L2 cache and centralized crossbar

already presents problems for today's processors. Consider, for example, the two paths highlighted in Figure 1. One path of approximately 15mm joins the upper-left core to the lower right one, while another path of approximately 12mm joins the upper-right core to a cache array immediately next to it. Notice that the path lengths correspond only weakly to the actual distances between the endpoints: In particular, the cache request of the upper right core becomes significantly slower and more energy consuming due to the detour through the central crossbar.

While it would be preferable to have a more efficient solution to interconnecting processor areas today, it will be *required* to interconnect the hundreds of processor cores of future high-performance processors. Ideally, one would like the cost (in terms of time and energy) of transporting data within and between the memories and processor cores of a supercomputer (whether on the same chip or on different chips) to be limited by the fundamental physical costs of such data transport.

Perhaps the most straightforward way to do this is to add a complete, circuit-switched and/or packet routed *network* to each chip that requires it. This technique of adding so-called "networks on-chip" (NoCs) [1] is helpful in a number of ways.

Firstly, they allow processor designers to avoid the problems associated with wire scaling in modern (and future) VLSI processes. For a number of ultimately economic rea-

sons, the physical size of a high-performance processor is unlikely to exceed 400mm^2 for the foreseeable future. However, the size of not only the transistors on the processor, but also of the *wires* interconnecting them, will continue to decrease. In particular, as wires approach fundamental physical limits (e.g. resistances dominated by the effect their cross-sectional area has on the electronic mean free path), it will become increasingly important for processor designers to avoid long on-chip wires wherever possible. One of the most promising ways they will be able to do this is to use NoCs: for example, embedding a simple two-dimensional mesh network in a processor chip greatly reduces the maximum wire length.

Another reason NoCs are helpful is that they can make high-performance processors easier to design. Considering again the example of a mesh NoC, its uniformity means that the potentially difficult task of designing the NoC nodes only has to be accomplished *once*, since the network nodes, like the processor cores, may then be replicated (perhaps with some small transformations) throughout the chip.

3. AN EFFICIENT ON-CHIP NETWORK

So that future high-performance processors may operate in a more power-efficient manner, we have been developing efficient NoCs which meet three key criteria:

- It must be able to transmit information between points rapidly and with minimal energy consumption
- It must be able to route information as required without incurring delays, retries or otherwise interfering with the performance characterisitics of the other chip components
- It must be small, reliable, and easy to integrate with other chip blocks.

3.1 Point-to-point NoC links

We have developed the most energy-efficient high-speed NoC links known, able to transport information across 4.5mm distances at a rate of 8GHz while consuming less than 50fJ per bit, several times more efficient than previously published designs and more than an order of magnitude more efficient than the conventional technique of inserting appropriately sized and spaced chains of full-swing inverters.[2]

These links operate by using a capacitively-coupled transmitter to obtain pre-emphasis (compensating for the high capacitance of the long 700nm pitch NoC link wires) and low-swing, saving power while consuming a minimal amount of chip area.[3] A 16:12 compressor further decreases transmitter power by approximately 1/3 while also reducing the length of highly capacitive 200nm pitch crossbar region that must be traversed at each NoC node.

3.2 Efficient NoC Routing

Our NoCs are an effectively bufferless, hot-potato routed network.[4] The elimination of buffers significantly reduces latency, chip area and energy consumption, but comes at the expense of an increased cost of router control and an inability to make sophisticated routing decisions. However, we can amortize these costs by routing only time critical and/or control packets in this hot-potato manner. As these packets propagate through the NoC, crossbar ports at each network node are reserved. Once a packet has reached its destination, a circuit-switched path has been established and may be used as if it were a point-to-point bus made of physical wires.

While this technique appears to involve additional latency (in that the transmitter cannot begin sending data until an acknowledgement is received that the full end-to-end path has been formed), in practice this penalty is insignificant as the overwhelming majority of NoC traffic in a scientific processor context consists of either long-lived "streaming" data flows or latency-critical cache misses in which case the transmitter has nothing to send, and must in any case wait for a response from the remote cache array that will satisfy the request (or in turn miss to the next level of memory).

In practice, our NoC implemented in a 400mm^2, 64-core processor can support random cache requests with an average latency penalty of only 720ps beyond optimally spaced point-to-point links, or 3 cycles at 4GHz, even while any 16 cores permanently occupy connections to any other 16 cores. NoC saturation occurs only when the NoC is approximately 65% loaded, due in large part to a folded torus-like network topology which features links of differing lengths, allowing routes to be detoured while still making forward progress in one or both axes.

3.3 Reliability and Ease of Integration

The elimination of buffers means that data is held in either the network link lines, or outside the NoC proper in other processor structures such as cache arrays or register files. This increases the reliability of the NoC with respect to single-event upsets (SEUs), since a single ionizing event will have a neglible effect on a highly capacitive and physically extended link wire, and should be unlikely to corrupt critical data in a cache array or register file which is properly designed to survive SEUs.

Regardless of the performance and reliability of an NoC design, it is unlikely to be adopted by conservative processor vendors unless it is easy to integrate (and in turn, evalulate). Therefore, our NoC has been carefully developed to integrate with standard cell-based design tools and techniques. In particular, all sections of the NoC, including highly sensitive analog sections (such as receiver-side sense amplifiers) are implemented as blocks that can be placed either as macro blocks or as extended-height standard cells. Since the link and crossbar wires must be carefully routed, this routing is performed by a custom router which can parse LEF/DEF/GDS and route around various chip blockages such as clock and power grids or memory macros.

4. REFERENCES

[1] A. Jantsch and H. Tenhunen (eds.) Networks on Chip *Kluwer Academic Publishers, 2003*

[2] D. Madina and M. Taiji. Circuit and physical design of the MDGRAPE-4 on-chip network links. In *Proc. SLIP* pp. 59-64, 2008.

[3] E. Mensink, D. Schinkel, E. A. M. Klumperink, A. J. M. van Tuijl, and B. Nauta. A 0.28pj/b 2gb/s/ch transceiver in 90nm cmos for 10mm on-chip interconnects. In *Proc. IEEE ISSCC*, pp. 414-415, 2007.

[4] C. Busch, M. Herlihy and R. Wattenhofer. Routing without Flow Control. In *Proc. SPAA*, pp. 11-20, 2001.