Impact of the Interconnect on Performance and Area/Power for High Core Count (> 8) CMPs

David Soper and Roy Zeighami

EE382C Spring 2007

Introduction

Faithful to Moore's law, silicon processing improvements have continually increased the number of transistors available for implementing CPUs within a fixed die area. The designer is left with the choice of how to put those transistors to use. Superscalar processors are organized into parallel pipelines which aggressively seek to execute instructions within a single thread in parallel. Unfortunately, this approach does not scale well, as a typical program would not have more than a few instructions that can be executed in parallel. An alternate approach has lead to processor designs that have multiple processing cores on each chip. Several chip multiprocessors (CMPs) are commercially available today (IBM's Power5 and Intel's Core Duo for example). Currently available designs put a small number of cores on each chip (2 to 4), but future designs will likely include many more cores on each chip. CMPs have the benefit of putting multiple cores on a single piece of silicon, so they can possible share a common L2 cache and can communicate without have to transmit data through an chip to PCB interconnect.

As the number of cores increases, chip designers must find ways to provide a low latency, high bandwidth interconnect between the processing cores. The on chip interconnect that a CMP uses has different design goals and constraints than a typical inter-chip interconnect. Since each chip will be a multiprocessor, the design should provide a high performance interconnect that can support the communication needs of the cores. The small dimensions of an on chip interconnect create the opportunity for higher performance than would be found in a multi-chip multi-processor (assuming roughly the same per core performance).

Because of constraints on chip area and power, the design of a high core count CMP must be aware of processor core and interconnect area and power costs on the CMP. The area and power constraints of the chip interconnect are more important to the design than they would be for an inter-chip interconnect. Power also plays a prominent role in the physical design of the interconnect, as a significant amount of the power dissipated in a CMP occurs in the wires.

In this paper, we review several papers that look at future CMP designs. Each paper focuses on the on-chip interconnect and how it affects performance, area, and power of the CMP as a whole. Since larger interconnects provide more challenges for performance and area/power, we focus on analysis of future CMP designs with high cores per chip (> 8 cores per chip). There currently does not seem to be a great deal of research that

analyzes the interconnect for large core count CMPs, but it is an area that should see increased attention over the next few years.

The papers we have chosen each analyze the interconnect of a proposed CMP, but they each analyze it in a slightly different way. We believe that together these papers give a good idea of how to analyze a high core count CMP and predict it's feasibility in terms of performance, total area, and total power consumption. Many papers study one aspect of a CMP, but few look at all aspects of the design. Without looking at all impacts of the interconnect on the design, high core count CMPs could prove difficult to actually build.

Future High Core Count CMP Interconnects

As core counts of CMPs increase, the interconnect plays an increasing role in the CMP design. The on chip interconnect for a CMP must allow for good performance on whatever workload the CMP is being designed for, and it also must fit within area/power constraints of the chip. With high core counts, the interconnect may take a large portion of the total chip area which will reduce the area available for processing cores. The designer may then have to make tradeoffs in the design, such as reducing the core count on the chip or using smaller cores which may provide less per core performance or less per core memory. The interconnect may also consume a significant portion of the CMPs power budget which again may force the designer to reduce power available for the cores.

There are many research papers covering many areas of CMP design. However, most focus on current and near term designs that only go to 8 cores on a chip. Two of the commonly cited papers covering 8 core CMP design are Compaq's Piranha paper [1] and Sun's Niagara [2]. Both of these papers describe the overall design of the CMP and give detailed performance estimates. These papers briefly mention the interconnects used on the chip for each design, but the interconnect is not a focus since it is a relatively small portion of the overall design. We focused on research covering future CMPs where the interconnect plays a more important role. We found only a handful of papers that mention core counts beyond 8, but there are already plans announced by companies like Intel to build high core count CMPs (Intel's Terascale project discusses an 80 core CMP [3]).

Below we review papers that focus on total chip and chip interconnect design. We found it difficult to find papers that discussed future CMP interconnects in detail, but we feel that the papers below lay out an excellent framework for evaluating a high core count CMP's overall design. A CMP will be designed with some workload (or set of workloads in mind), so the designer must be careful to keep the workload in mind when deciding what core design will be used.

High performance, out-of-order execution cores have typically been used for server workloads (such as on-line transaction processing, or OLTP). These server workloads typically have high demands on the memory subsystem and require large, private caches (these workloads are said to be cache or memory bound [4]). If a CMP is designed for server workloads and the designer favors complex cores with large per core memory, the

number of cores allowed on a chip and the area/power budget for the interconnect may be limited. Alternative designs, like those for desktop computing, which has few parallel applications, may call for less complex, in-order execution cores with smaller per core memory. These designs may leave more room and power for the interconnect and allow for many more cores on the chip. In any case, approaching the core and interconnect design together and being mindful of the expected workload is very important when designing a high core count CMP.

Review of "Design Tradeoffs for Tiled CMP On-Chip Networks" [5]

Area of focus

This paper focuses on analysis of performance and of area/energy efficiency on a 64 core tiled CMP. Such a high core count design has constraints on the interconnect, especially in terms of chip area and chip power dissipation. Wire delay of the interconnect is shown to factor more into the overall performance of the interconnect. There is likely much more communication between cores with a large design so providing good performance while keeping area and power in mind is important. This paper looks at various topologies for connecting cores within the CMP and asks the question of how the performance changes if the network is duplicated. Also, the question of how to split the messages on the network is addressed.

Evaluation methods

The analysis looks at the whole on-chip network including the network's topology, routing, and flow control. To perform an analysis of a variety of networks and a variety of traffic patterns, detailed area and energy models are developed for on chip routers and the interconnect wiring. Models of all components of the network are created. These models are based on a 65 nm process with 8 total metal layers (4 available for interconnect wiring).

The interconnect components (wires and routers) are basically fixed for the analysis, but topologies are changed in order to measure the topology's affect on performance and cost in terms of area/power. Each network evaluated consists of 64 processor tiles and each tile has a processing unit with 256 KB of local memory. The processing models aren't the focus of the paper so the models appear fairly simple and a fully cache coherent memory hierarchy is not modeled.

When analyzing the impact of the interconnect, a primary focus is on the router. To analyze the router, models for the flit buffers (in SRAM), input logic, virtual channel (VC) allocator, and the router switch are created. For all topologies, virtual channel routers are used. The router model is fairly complex and each router uses route lookahead, speculative virtual channel allocation, and speculative switch allocation. The fast route lookup and speculation allows the typical router traversal time to be only 2 cycles.

Since the routers consume a significant portion of the area on the chip, all parts of the router are modeled in detail. The input module (with virtual channel state and input buffering) and its buffer storage elements are well designed to reduce power consumption and provide for fast lookup (which reduces latency). Both short and long flits (used for different request/reply types) can be accessed in a single cycle. Switch and output module design is also meant to reduce latency while using minimum power. Channel design (wire length and repeater layout) attempts to again minimize power and latency. The design is constrained by the allowed cycle time (packets can't arrive off cycle at downstream routers).

With area and energy models created for the routers and their channels, the evaluation focuses on network topologies and how they perform relative to their area and energy costs. Several direct (torus/mesh) networks are considered along with several indirect (butterfly) networks. The direct topologies used include an 8-ary 2-cube mesh and torus and a 4-ary 2-cube concentrated mesh (4 nodes concentrate traffic onto 1 router). The indirect topologies used are both versions of folded Clos. Both indirect topologies are forms of "fat-trees". The tapered fat-tree that is described reduces bandwidth near the root of the network since less traffic should reach the root than reaches other switching nodes.

For each topology, an optimal routing and flow control technique is used. The routing and flow control techniques used should provide the best performance across a variety of traffic patterns for each network. For example, the fat-tree uses an adaptive algorithm to route to a nearest ancestor then route to the destination (a form of minimal adaptive routing that should perform well on any traffic pattern). One interesting note is that the 2 meshes evaluated (regular mesh and concentrated mesh) both leave enough area on the chip for a parallel network.

Results

The network models described above are used to measure how each network design affects the area and power used by the chip. For each configuration, various parameters (datapath width, buffer depths, etc.) are varied in simulation to determine the optimal configuration. Once the optimal parameters are established, simulations are run on all the networks to compare performance. The analysis of each network's area and energy efficiency is done using a variety of traffic patterns.

From these models and the traffic simulations, area-efficiency and energy-efficiency is measured. The area and energy efficiency is a measure of the network's performance relative to its area and energy consumption. More specifically, the product of the workload completion time and area or energy dissipated is used for each comparison. The optimal network configurations are those that maximize performance (short completion time) while minimizing area and energy use.

Results show that a concentrated mesh is the overall best design and the area and power use is even low enough that a 2nd parallel network can be used. Further, the preferred approach was to use one of the network for write requests and replies and the other for

read request/replies. This is in contrast to the less successful approach of using one network for long packets and the other for short messages. The latter approact left the network with short messages underutilized. Somewhat surprisingly, mesh networks (which are often proposed for tiled interconnect) have the worst efficiencies. The mesh is the poorest performer (long completion times and high average latency) by over 25% which leads to poor efficiencies.

Critique

Analyzing the area and energy efficiency of each possible interconnect provides a good way to measure the effectiveness of a design. The efficiencies measures allow for good comparisons between an interconnect's performance and its cost in terms of area/power. With increasing transistor counts per unit area, power dissipation becomes more of a concern However, many papers fail to mention area or power analysis when evaluating the design of a CMP, so we see this analysis as a strength of this paper.

The models of the interconnect and its routers are straightforward enough to allow for good evaluation without spending too much time focused on circuit design, yet they still are detailed enough for accurate simulation. These models also covered all important parts of the network. Such detailed modeling was not found in other papers we read.

While the models developed provide a good framework for evaluating the interconnect of a CMP, the traffic patterns and cpu models used are not a focus of the evaluation. Since a particular workload (or class of workloads) wasn't really targeted by the paper, it seems reasonable to think that this particular CMP design might not necessarily be a good fit for a particular workload. For example, if the proposed CMP design was used in a commercial server running an OLTP application, the memory demands of the workload might not be met. Other interconnect designs (besides the concentrated mesh) might be better suited to certain workloads since those might generate alternate traffic patterns. For certain workloads, the high raw performance of other topologies might be worth the higher area and/or energy costs.

This papers answers the specific question of "holding all things constant what is the best topology for interconnting a 64 way CMP?" However, it leaves certain other questions unanswered. For example, what if the two networks in the CMESHX2 topology were not identical? Would the long message/short messaging partitioning have worked better if the two network have diffent channel widths? There seems to be some opportunity for someone to extend the work carried in this paper to examine some of these other interesting questions.

The next paper we evaluate [6], does not provide the same in depth coverage of the interconnect but does look more closely at core and memory demands for certain workloads. Using real benchmark applications in evaluation to estimate performance for server and/or desktop workloads seems like a good next step when evaluating the proposed design. We did feel that this paper did a good job focusing on the interconnect and it provides excellent techniques for evaluating the interconnect as part of a CMP design. However, for actual designs, it seems like a good idea to use the analysis ideas

covered in the paper along with actual workloads to get a feel for how the whole CMP would behave with real applications.

Review of "Interconnects in Multi-core Architectures" [6]

Problem Addressed

This paper was in some ways similar to [5] and again focused on performance and analysis of area/power efficiency on a high core count CMP. A key difference is that this paper focused on the core processor design (per core performance and memory) and the workload expected to run on the CMP. Designing the core and interconnect together is considered essential when designing a CMP. If the interconnect is designed independent of the core, a high bandwidth interconnect might not leave enough area/power for cores that meet the need of the applications. Likewise, large caches without the bandwidth to interface to them is equally problematic.

The primary goals of the paper are to develop detailed models to carry out a performance analysis, to compare interconnect topologies for 8-16 core CMPs, and to show how processor and memory design is important along with interconnect. Also, the paper explores tradeoffs within the memory hierarchy of CMPs. In particular, the tradeoff between a shared L2 cache or private L2 caches.

Evaluation Methods

Again, latency, power, and area are the focus of the evaluation. A 65nm process with 10 metal layers is considered. Wiring and logic models are developed to use in the evaluation of several topologies. The topologies considered include a bus based interconnect supporting up to 8 cores. This "shared bus fabric" is a high capacity bus meant to support the high traffic demands of a high performance out-of-order execution processor with a large private cache. Significant wiring and logic is needed for each bus interconnect. To support high core counts, multiple shared bus fabrics can be connected through a point-to-point link. This allows modeling of up to 16 cores on a single chip. To study the performance of sharing caches, a crossbar interconnect is also modeled. The crossbar would allow simultaneous access to shared caches from several processors. The bus and wiring models are explained in detail and area and power costs for each component (control logic, bus queues, and wiring) are calculated.

Simulation of real world workloads is a focus of the paper. A full coherence protocol (MESI based) is simulated and models are validated against implemented designs. Workloads include commercial benchmarks such as TPC-C (for on-line transaction processing).

Results

Data from shipping systems and a variety of commercial workloads is used when simulating performance on the proposed topologies. For each topology, performance is measured and compared against area and power consumption. 4, 8, and 16 cores are evaluated on each of the interconnects (shared bus fabric, shared busses with point-to-point between bus connections, and the crossbar).

For the basic shared bus, an initial configuration with 3MB per core caches for 8 cores and 0.5 MB caches for 16 cores is used. In the initial configuration, area and power consumption by the bus interconnect is significant. The 16 core interconnect occupies 13% of the total chip area and power increases superlinearly with core count.

Wire count and length must both increase as more cores are connected to the bus. The area and power impacts of the wiring and control logic for the bus is significant, but there are ways to reduce their impacts while not impacting performance much. As bus interconnect impact is reduced (by reducing width and/or bus frequency), the saved area and power can be put into the caches which can actually boost performance. Simulation results show that slight reductions to bus size and frequency actually provide the best performance because of larger caches. However, decreasing bus bandwidth too much actually begins to hurt performance. The results show the importance of designing processor/memory hierarchies along with the interconnect for best area and power efficiency.

This paper refutes the conventional wisdom that a large shared L2 caches are a foregone conclusion for CMPs. Standard thinking says that a shared cache is better than private L2 caches because shared caches lines need not be repeated, hence the effective size of the L2 is increased. However, the authors measure the CPI for four levels of cache sharing: all_private, two_shared, four_shared, and all_shared. When no interconnect overhead is assumed, performance increases with increased sharing, as expected. However, when overhead is factored in, the minimum CPI is achieved at less than full sharing. Further, the actual performance depends heavily on the actual metal layer used. The upshot is that some sharing seems to improve performance but too much sharing burdens the interconnect and hurts performance.

The workloads place heavy demands on the interconnect because they are all memory intensive. With a large, shared cache, the interconnect must be able to support a large amount of traffic generated by all processors. The crossbar interconnect cost (in terms of area and power) is very significant for the workloads studied and the paper concludes that cache sharing is not a sensible option for commercial server workloads with their crossbar interconnect design.

Critique

The performance modeling is a strength of the paper. The use of detailed coherence models (with data from real systems) along with actual server workloads helps the paper give performance estimates of each possible design that should closely match what a production system would see. Commercial workload memory demands drive most of the overall design. Commercial workloads have poor cache locality when compared to other workloads. They also have high demands on the off core (or between core) memory subsystem. In other words, these workloads require a low cache miss penalty. The paper asserts that large caches are needed for reasonable performance. The interconnect and core design must be careful not to constrain cache size too much.

While the application modeling is interesting, the paper's main focus is on bus based interconnects, which was not an ideal topic for our research. Although the bus is a high bandwidth interconnect, it will not scale to high core counts (beyond 16 cores). Even at smaller core counts (8-16 cores is the paper's main focus), the wiring overhead of the bus is substantial. Also, the bus queues and control logic place high demands on the total CMP area and power budget. However, the analysis of the interconnect along with the workload performance is a novel approach and would apply well to any CMP interconnect design.

We found it interesting that the crossbar interconnect results are counter to those given in the Piranha paper[1]. Piranha is designed from the ground up to provide needed memory latency and bandwidth with shared caches. The interconnect and coherence protocol design are meant to satisfy the needs of high memory demand commercial server workloads. In this newer paper, the bus based interconnect (which is even used in the crossbar for coherency support) was likely not designed for shared memory type environments so we wonder if other interconnects using shared memory would lend themselves to server workloads.

Future work in this area might include a similar analysis done for higher core count CMPs using other interconnections that scale better. A crossbar based interconnect is considered in this paper, but only for the case of the cores sharing caches. A coherency protocol and workload modeling setup similar to what's described here along with an interconnect analysis like that done in [5] seems like it would be an excellent approach for determining the expected performance of a large CMP on real workloads.

Review of "Interconnect-Aware Coherence Protocols for Chip Multiprocessors" [7]

Area of focus

Parallel workloads in CMPs typically have high demands on the memory system (the memory and interconnect beyond private caches that must service cache misses). The workloads have high private cache miss rates (L1 miss rates in a design with private L1s and shared L2s) that result in considerable memory traffic. The memory system must support low latency and high bandwidth memory accesses and must support the overheads of the coherency protocol which must be used to keep private L1s coherent. There are generally two options for coherency, a snoopy bus based protocol (where all bus agents snoop all bus accesses and react accordingly) and a directory based protocol where the agent hosting memory must check cache line state on misses and make explicit requests of caching agents.

With any coherence approach, the demands of the interconnect must be considered when designing the system. Wire delay and power consumption of the interconnect will have significant affects on communication between processors and their caches on the CMP. Wiring choices affect communication latency and bandwidth to a large degree, and should be considered by the system architect instead of only by the VLSI/circuit designers.

Since different communication within the network has different requirements, it may be valuable to design different parts of the interconnect (and specifically the interconnect wiring) in different ways. For example, different wiring might be used to service short data transfers with low latency requirements and longer data transfers that require more bandwidth but can tolerate higher latency.

The performance of interconnect wiring (in terms of wire delay) is primarily affected by the width and spacing of the wires. Wires that are wide and far apart provide the lowest latency, but fewer wires are allowed per unit area so bandwidth is reduced. The length of wires and their delay constraints affects the power they consume. Long wires that must minimize delay require repeaters that consume power. Power consumption can be reduced with shorter wires with higher latency. Rather than design the whole system using the same wiring scheme, the paper proposes 3 wire types:

B-wires – standard wires
L-wires – low latency wires with reduced bandwidth (1/2 the latency of B wires)
PW-wires – low power wires with increased latency (2x the latency of B wires)

Further complicating the design proposed in this paper is the authors' choice to map messages to the different wire types dynamically. In other words, rather than map all messages of a single type to a certain type of wire, a runtime decision is made for routing certain messages depending on the state of the system. For example, the L2 cache may response to a read-exclusive message by transmitting on the B-wires- or PW-wires depending on whether there are other people sharing the cache line.

In an interconnect design, non-performance critical signals can use PW-wires, latency critical signals can use L-wires, and wires used for bulk data transfer can use B-wires.

Because different wires are used in different parts of the design and for different transaction types using the same channels, some overhead in router logic and buffering is introduced. For certain transactions, different wires must be used so there is additional mux/demux circuitry throughout the design (for example to mux certain transactions from a processor onto the preferred wire type). The routers must have additional virtual channels (and their state/buffer resources for each wire type). Additionally, latches must be placed throughout the interconnect to allow different wire speeds to be synchronized against the global interconnect clock.

Evaluation methods

Performance simulations model a 16 core CMP on a 65 nm process with 10 metal layers. Both in-order and out-of-order processor cores are simulated across the set of SPLASH-2 application benchmarks (a variety of technical computing benchmarks). One focus of the paper is on how coherency traffic affects the interconnect, so realistic coherency models are used in all the simulations. A directory based coherence scheme (in which the L2 must maintain coherence state of all cache lines) is used and optimized to use different

wire types for various operations (low latency wires when querying the L2 for directory state as an example).

A 2-level tree interconnect (4-ary 2-fly) and 2D (4-ary 2-cube) torus are both evaluated. Both interconnects use adaptive routing and virtual channel flow control. Performance is evaluated relative to a design using only one wire model. Power is also measured for each configuration and workload and compared with the homogeneous wire model.

Results

Results show that on average using the multiple wire (heterogeneous) types provides about an 11% improvement in performance and a 20%+ reduction in power. The results are quite thorough and also show how the heterogeneous model is not ideal for all designs (the performance actually decreases in a bandwidth constrained system without abundant wiring). In order and out-of-order processor cores are also compared. The out-of-order cores actually show a smaller performance gain when compared to a homogeneous wiring model since the out-of-order cores are less sensitive to latency (less affected by latency differences in the non latency optimized configuration).

Critique

As the other papers showed, the interconnect wiring has significant impact on the performance and power cost of an on-chip network. For parallel workloads, a significant amount of network traffic is generated by each processor and designing the interconnect and its wiring to meet the demand is essential. The focus of this paper is on the coherence traffic of a CMP was an interesting approach. As the paper points out, coherence traffic has several different requirements (low latency vs. high bandwidth for example), so there's a good opportunity to tune the network for each traffic type.

The paper did a good job of considering a variety a coherence schemes, interconnect topologies, and processor models (in-order vs. out-of-order). Although the paper's focus was on how the interconnect affected coherence operations, the modeling done and performance/power analysis was similar to the other papers. An additional strength of this paper was that it combined some strengths of the other two (namely real workload simulation along with an analysis of scalable interconnect designs). The use of real benchmarks and detailed coherence models for memory should provide for accurate predictions of real world performance for this type of design. This paper also developed good models of the interconnect (including routers and wiring) and of the processors and their memory hierarchy. The use of different wiring types was very novel and we felt the evaluation results showed the real benefits this type of approach might have.

Future research in this area might focus on large CMPs and additional network topologies. A 16 core CMP was the focus of this paper, but the networks evaluated should be fairly scalable. A challenge as the design scales would be how to overcome increasing latency of coherence communication as distance between cores increases.

References

1. L. Barroso, K. Gharachorloo, R. McNamara, A. Nowatzyk, S. Qadeer, B. Sano, S. Smith, R. Stets, and B. Verghese. *Piranha: A Scalable ArchitectureBased on Single-Chip Multiprocessing*. Proceedings of the 27th Annual International Symposium on Computer Architecture. June 2000.
2. P. Kongetira, K. Aingaran, K. Olukotun. *NIAGARA: A 32-WAY MULTITHREADED SPARC PROCESSOR*. IEEE Computer Society. April 2005.
3. J. Held, J. Bautista, S. Koehl. *From a Few Cores to Many: A Tera-scale Computing Research Overview*. Intel White Paper. 2006.
4. J. Huh, S. Keckler, and D. Burger. *Exploring the Design Space of Future CMPs.* Proceedings of the 2001 International Conference on Parallel Architecture and Compilation Techniques. 2001.
5. J. Balfour and W. Dally. *Design Tradeoffs for Tiled CMP On-Chip Networks.* ACM. 2006.
6. R. Kumar, V. Zyuban, and D. Tullsen. *Interconnections in Multi-core Architectures: Understanding Mechanisms, Overheads, and Scaling.* IEEE. 2005.
7. L. Cheng, N. Muralimanohar, K. Ramani, R. Balasubramonian, and J. Carter. *Interconnect-Aware Coherence Protocols for Chip Multiprocessors.* School of Computing, University of Utah.