Published in IET Computers & Digital Techniques
Received on 4th July 2008
Revised on 18th December 2008
doi: 10.1049/iet-cdt.2008.0078

In Special Issue on Networks on Chip

ISSN 1751-8601

Impact of on-chip network parameters on NUCA cache performances

A. Bardine M. Comparetti P. Foglia G. Gabrielli C.A. Prete

Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Via Diotisalvi 2, 56122 Pisa, Italy
All authors are members of the European HiPEAC Network of Excellence
E-mail: alessandro.bardine@iet.unipi.it

Abstract: Non-uniform cache architectures (NUCAs) are a novel design paradigm for large last-level on-chip caches, which have been introduced to deliver low access latencies in wire-delay-dominated environments. Their structure is partitioned into sub-banks and the resulting access latency is a function of the physical position of the requested data. Typically, NUCA caches employ a switched network, made up of links and routers with buffered queues, to connect the different sub-banks and the cache controller, and the characteristics of the network elements may affect the performance of the entire system. This work analyses how different parameters for the network routers, namely cut-through latency and buffering capacity, affect the overall performance of NUCA-based systems for the single processor case, assuming a reference NUCA organisation proposed in literature. The entire analysis is performed utilising a cycle-accurate execution-driven simulator of the entire system and real workloads. The results indicate that the sensitivity of the system to the cut-through latency is very high, thus limiting the effectiveness of the NUCA solution, and that modest buffering capacity is sufficient to achieve a good performance level. As a consequence, in this work we propose an alternative clustered NUCA organisation that limits the average number of hops experienced by cache accesses. This organisation is better performing and scales better as the cut-through latency increases, thus simplifying the implementation of routers, and it is also more effective than another latency reduction solution proposed in literature (hybrid network).

1 Introduction

Non-uniform cache architectures (NUCAs) have been proposed as a novel design paradigm for large last-level on-chip caches [1] in order to reduce the effects of wire delays, which significantly limit the performance scaling of today's high clock frequency microprocessors [2]. This is achieved by the adoption of a storage structure partitioned into sub-banks, with each sub-bank being an independently accessible entity, and by the adoption of a fast interconnection network to connect the banks and the cache controller. The access latency exhibited by a NUCA cache is a function of the physical location of the requested line, for example a line belonging to a bank located near the cache controller will be accessed faster than another line belonging to a bank located farther away. The mapping between cache lines and banks can be either static or dynamic. The former approach leads to the static NUCA (S-NUCA) scheme: a line can be located in a single specific bank, univocally determined by its address. The latter approach leads to the dynamic NUCA (D-NUCA) scheme: a line can be located in one of a set of allowed bank locations, which collectively form a bank set, and each bank of the bank set behaves like a single way of a set-associative cache [1]. Lines can dynamically migrate from one bank to another, provided that it belongs to the pertaining bank set, and the migration is triggered by a certain number of consecutive line accesses.

A viable solution to connect the banks and the controller of a NUCA cache is represented by an on-chip network [3, 4]. The paradigm introduced by on-chip networks tends to favour the reuse of design and verification efforts, which is

particularly important for modern VLSI processes: many digital design blocks, namely network links and routers, can be used repeatedly to form a complete communication infrastructure across the chip. The resulting interconnection scheme is more scalable than traditional approaches based on broadcast media, such as busses and rings. The intrinsic features of NUCA caches introduce constraints on the design of the on-chip network, in particular on the design of the network routers. These constraints impact the characteristics of the network itself, such as topology, routing and flow control, but, primarily, they are influenced by the way with which last-level on-chip caches are accessed by the CPU. A fundamental property of the NUCA on-chip network is that it is self-throttling [5], as it is common for processor-to-memory interconnects. In fact, non-blocking caches are able to support only a limited number of outstanding misses, therefore the number of simultaneous requests on the L2 or, more generally, on the last-level cache, is limited by the number of outstanding misses supported by the higher level. This number is determined by the number and size of the Miss Status Holding Registers (MSHRs) [6], which are used to keep track of the pending misses, coalescing multiple outstanding misses for the same cache line into a single request to the following level of the memory hierarchy. From these considerations, we might expect the network traffic offered to the NUCA on-chip network to be quite moderate. Since the access latency is the fundamental performance metric of a NUCA cache, together with the hit rate, we also might expect latency, instead of bandwidth, to be the primary design goal for the switching elements of the network, in order to build fast NUCA caches.

Different implementations of routers have been proposed for high-performance on-chip networks, but it is not clear which is the most suitable router architecture to face the design constraints posed by a NUCA cache scenario. In order to characterise such design constraints, in this paper we analyse how the performance of a reference NUCA L2 cache [1, 7, 8] is influenced by different values of cut-through latency and buffering capacity of routers. Such parameters, in fact, allow to guide the selection of an adequate router architecture. The entire analysis is performed with a cycle-accurate execution-driven simulation platform that is able to precisely model the CPU and the memory hierarchy, including all the aspects regarding the interconnection network employed for the NUCA L2 cache, and a preliminary design space exploration has been performed to identify the reference architecture parameters. The obtained results show that different implementations of network routers can significantly affect the overall system performance; in particular, the sensitivity to the cut-through latency is very high, while varying the amount of buffering resources has small effects and single-message sized buffers enable to achieve adequate performance levels. Taking into account these considerations, we propose an alternative NUCA organisation based on bank clustering which is able to reduce the strong sensitivity to the cut-through latency of routers, thus enabling higher performance levels and simplifying the implementation of the routers. Our technique is also better performing than another latency reduction solution proposed in literature (hybrid network).

2 Related work

The NUCA cache paradigm has been introduced by Kim *et al.* [1] for the single processor case. Chishti *et al.* [9] have proposed an optimisation scheme, called NuRAPID, aiming at increasing the energy efficiency and the performance of NUCA caches in a single-core configuration. Several studies have focused on the application of NUCA caches to CMP (Chip MultiProcessor) architectures, focusing on the evaluation of the best sharing degree [10], on the effectiveness of block migration for multithreaded workloads [11] and on optimisations for block placement and coherence management [12]. Nevertheless, none of the studies described so far has explicitly focused on the impact of the network architecture on the overall system performance.

Other studies have focused on different design aspects of the on-chip network for NUCA caches, such as the one by Muralimanohar *et al.* [13]. In their work, they have introduced the ability to model S-NUCA caches within the CACTI tool (CACTI 6): the design space exploration has been augmented with different wire types and includes on-chip network parameters for NUCA, such as the number of pipeline stages, the size of buffers and the number of virtual channels per router. However, the effects of varying these network parameters on the system performances have not been reported in their work. In addition, the evaluation methodology that we used in our analysis is quite different from the one described in [13]: while CACTI 6 determines the effects of the contention on the network resources according to a look-up table (which has been populated a priori through simulations for some representative workloads and for different processors counts), in our analysis we rely on a cycle-accurate simulation of the network, thus modelling with high precision all the dynamic effects due to the contention on the network resources and to the D-NUCA migration mechanism, against real workloads. In another work [14], Muralimanohar and Balasubramonian focus on the design of the interconnections between the banks and the cache controller(s) of an S-NUCA L2 cache, for single-processor and CMP systems. In particular, they investigate the performance improvements obtained by employing different interconnection schemes distributed on multiple metal layers, assuming a three-stage pipelined router with an unloaded latency of three cycles and virtual channel capabilities. In this sense, their study can be considered orthogonal to our analysis. Another optimisation presented in [14] leverages a hybrid approach to design the network topology that reduces the wiring resources by mixing

IET Comput. Digit. Tech., 2009, Vol. 3, Iss. 5, pp. 501–512
doi: 10.1049/iet-cdt.2008.0078

point-to-point links and busses (hybrid network). In Section 7, the evaluation of our proposed technique to reduce the impact of the network latency on the overall system performance includes a comparison with the hybrid network scheme, since both can be viewed as techniques to reduce the average number of hops and then to reduce the sensitivity to the routing latency. Jin *et al.* [15] have shown that the network traversal time is the main component of the latency of a NUCA cache access and they propose solutions based on data management, such as the block management policy called fast-LRU, which augments the D-NUCA migration mechanism in order to better approximate the LRU ordering of blocks with respect to the policies introduced in [1]. However, the overhead introduced by these solutions, especially in terms of extra energy consumption and network traffic, has not been fully characterised.

On-chip networks have been introduced as a common communication infrastructure for system-on-chips, which are an emerging paradigm for designing embedded and application-specific systems. In the field of general purpose high-performance systems, on-chip networks have been proposed for different purposes. A typical application is represented by tiled CMP architectures [16, 17], which are based on a matrix of nodes (called tiles), with each node comprising a processing unit and, in most of the cases, a certain amount of cache memory. An on-chip network connecting the tiles is responsible to transport the data and synchronisation messages, enabling chip-wide communications. Intel has adopted an on-chip network to connect the 80 cores of the tera-scale research project called Teraflops [18]. The TRIPS processor prototype [19] has employed different on-chip networks to interconnect the execution units, the SRAM cache banks of an S-NUCA L2 cache and the DRAM controllers; since the prototype has been built with a 130 nm manufacturing technology and its operating frequency is fixed at 500 MHz, the effects of wire delays encountered for that design are not so critical as in deep sub-micron manufacturing technologies, as pointed out by the same authors of [19]. The impact of router delay on the performance of a grid processor employing an inter-ALU operand network has been analysed in [20], and the results indicate that such network structures are highly sensitive to this parameter. Balfour and Dally [21] evaluate different topological alternatives of on-chip networks for tiled architectures. They propose a topology organisation, called CMesh, that is able to reduce the average access latency and it is particularly effective when the traffic exhibits frequent local patterns, which are common for typical workloads for this kind of architecture. While the CMesh is a topology similar to the clustered approach introduced in this paper, the results obtained by Balfour and Dally are not directly applicable to our case, because in the context of NUCA caches the traffic patterns between cache banks do not exist, except for the traffic because of the D-NUCA migration mechanism which is negligible with respect to the traffic between the cache controller and the cache banks.

The infrastructure of on-chip networks for general purpose systems must support high operating frequencies, as it is common for modern chips, and several works have focused on the design of high-performance network routers for this kind of applications. Most of them focuses on a pipelined organisation, which is able to guarantee a high throughput. For instance, Peh and Dally [22] propose a speculative router model with a three-stage pipelined architecture with virtual channels. Clearly, the main design objectives for high-performance routers are to minimise latency, by reducing the number of pipeline stages, and to maximise throughput; Mullins *et al.* [23] have proposed an innovative router architecture with virtual channels, which is able to deliver a flit in a single cycle, using speculation mechanisms, buffer bypassing capabilities and removing the arbitration logic from the critical path; Jin *et al.* [15] also adopt a single-cycle router with virtual channel capabilities, but its architectural implementation has not been detailed. However, while such architectures have been proposed in literature, single-stage routers are not yet an industrial reality [14] because of the issues that emerge in the digital design process. Therefore it is relevant to analyse the performance sensitivity of NUCA caches to the router delay, since designing low latency network routers is not a trivial task.

Concerning the buffer capacity of routers, it is important to analyse the buffering requirements of NUCA on-chip networks, since, apart from performance, the size of queues has a significant impact on dynamic and static energy consumption and on die area occupancy [24, 25].

3 On-chip network architecture and router model

The analysis described in this paper assumes a reference NUCA structure which has been derived from previous works [1, 7, 8]. The topology of the on-chip network is derived from a 2D mesh by employing only a subset of the links of a full 2D mesh in order to reduce the area overhead, resulting in a tree topology. Different topology schemes of direct networks, such as toruses, have not been considered for this study because 2D meshes and trees map more effectively onto 2D silicon substrates: even if a torus can be mapped onto 2D silicon substrates, doing so in a wire-delay-dominated environment would not introduce significant improvements, except for bandwidth gain, and will increase the area occupied by the interconnection fabric. The sole injection point of the network is the L2 cache controller, which is assumed to be directly attached to the external DRAM controller. The network links are bidirectional, so two traffic flows on opposite directions are completely independent from each other. The tree topology of the on-chip network is represented in Fig. 1.

The reference architecture for the NUCA on-chip network is based on a wormhole scheme, with routing and flow control

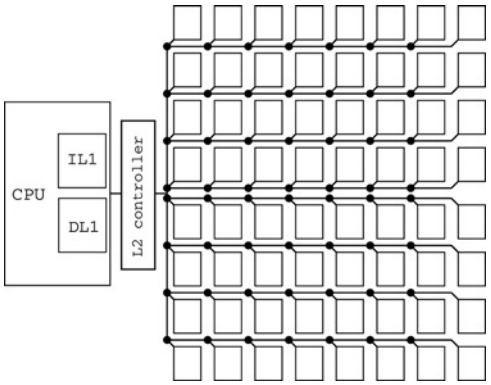

Figure 1 *Selected topology for NUCA architectures. The NUCA structure represented here is made up of 64 banks (8 × 8). The black circles depict the network routers. For this study, the same topological organisation is used for both S-NUCA and D-NUCA architectures. The CPU, with its instruction and data L1 caches, is attached through a bus to the L2 cache controller, which is the injection point of the NUCA on-chip network. For D-NUCA, each row of banks corresponds to a bank set*

policies working on a per-flit basis. The size of a flit is assumed to be equal to the link width. The routing scheme is deterministic, dimension ordered; for the NUCA architectures considered in this work, a flit belonging to a packet corresponding to a cache access request is first propagated along the vertical dimension (vertical links in Fig. 1), then it is propagated along the horizontal dimension (horizontal links in Fig. 1); flits belonging to reply packets follow the same path of the request packets. For D-NUCA caches, since a bank set is mapped to a single row of banks, first a flit has to reach the pertaining bank set, and then it is propagated to the nodes attached to the banks of its bank set, starting from the nearest one to the cache controller. Such a behaviour determines a global cache access latency that gets higher as the distance of the requested cache line from the first node of the pertaining bank set increases.

The general architecture of the network routers that have been modelled for this analysis is represented in Fig. 2. The network routers are assumed to be input buffered, and the buffers are managed on a per-flit basis in a FIFO manner. The flow control is credit based, so each router must keep track of the status of the input queues of its neighbours. This is accomplished by adding two extra credit signals to the link width (credit in and credit out in Fig. 2). When a router removes a flit from one of its queues, it asserts the corresponding credit out signal to notify the sender that more buffering space is available. On the other side, a sender is allowed to transmit a flit only if the number of collected credits is not null. In order to guarantee routing fairness, a round-robin scheme is used when multiple transmission requests for the same channel occur. In order to implement this feature, the router needs to keep track of the direction of the last packet sent for each output channel.

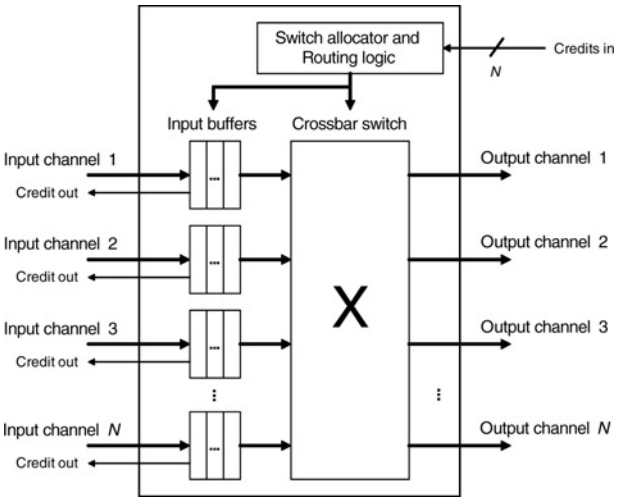

Figure 2 *Reference architecture for the network routers. The routers are assumed to be input buffered. A crossbar switch is adopted to minimise contention on output channels. The credit signals are necessary to implement credit-based flow control*

The variable router parameters that have been considered in this study are: (i) cut-through latency (expressed in number of clock cycles); (ii) buffer capacity (expressed in number of flits per input queue). The cut-through latency is equal to the delay needed to transfer a flit from the source input channel to the destination output channel of a router (Fig. 2), assuming a no load condition. In order to calculate the hop latency, that is the latency to move from one node to the next, our model takes the sum of the cut-through latency and the link latency (delay introduced by the transmission of signals on wires). The wire length has been determined assuming the full width or height of the cache banks, depending on the link direction (horizontal or vertical). The methodology that has been applied to calculate the physical parameters is described in the next section.

4 Physical parameters and system configurations

Computer architects often rely on existing analytical models to estimate the characteristics of VLSI circuits. One of such models has been adopted by the CACTI tool [26] to estimate the area, the access time and the energy consumption of on-chip SRAM caches. The analysis described in this paper is based on values obtained from CACTI 5.1 [27], which derives the technological parameters for devices and wires from the projections of the ITRS report [28].

Our analysis is based on a baseline system configuration comprising a single CPU with splitted instruction and data L1 caches and a unified L2 cache, whose architecture is varied between traditional UCA (uniform cache architecture), S-NUCA and D-NUCA. For each of the considered L2 cache architectures, we selected the best performing configuration, assuming a fixed capacity of

doi: 10.1049/iet-cdt.2008.0078

8 MB and a line size of 64 B. For UCA, we collected the simulation results for different configurations obtained by varying the cache associativity, from direct mapped to 32-way set associative. We used CACTI to determine the cache access time and cycle time for each case. From these experiments, based on the simulation methodology described in the following section, we derived that the maximum performance level is achieved for 16-way set associativity, but a four-way set associative configuration is able to achieve a performance level that differs from the maximum value by 0.3%. Since increasing the associativity of the cache always implies an increase of the design complexity and of the energy consumption, we chose four-way set associativity as the best tradeoff. For NUCA caches, the design space to be covered in order to identify the most performing configuration is wider. For S-NUCA, the parameters to be identified are the bank size, hence the total number of banks, their grouping into rows and columns, and the internal associativity of the banks. CACTI was used in this case to determine the access time, the cycle time and the physical height/width of the single banks, since they behave like traditional UCA caches. Regarding the bank associativity, we obtained the same results as for UCA and we selected four-way set associativity as the best tradeoff. Regarding the bank organisation, Fig. 3*a* reports the performance levels obtained for different bank configurations for S-NUCA: the highest average instructions per cycle (IPC) over the entire workload is achieved for the configuration with 32 banks, grouped into eight rows and four columns (8 × 4), but the levels achieved by the 16 × 4 (64 banks) and 16 × 8 (128 banks) configurations are very close; however, since a larger number of banks implies a significant increase of the design complexity, we selected the 8 × 4 configuration as the best one. Similarly, for D-NUCA we varied the bank size and the number of bank rows/columns, while the global degree of associativity, according to the selected D-NUCA implementation [1], is determined by the number of bank columns. Fig. 3*b* reports the performance levels obtained by the different D-NUCA bank organisations: the configuration with 64 banks, 8 × 8, globally behaving like an eight-way set associative cache, is evidently the best performing. Table 1 summarises the selected optimal configurations for UCA, S-NUCA and D-NUCA.

For the D-NUCA scheme, several implementation policies have been proposed [1]: mapping policies (simple, fair or shared mapping), line search policies (sequential or broadcast search) and migration policies (determined by the promotion trigger – number of accesses after which a line is promoted –, and by the promotion distance – number of banks traversed upon a promotion –). In order to keep the number of variable parameters reasonably low, in this study a specific set of policies has been selected for D-NUCA, leading to a configuration that is a good tradeoff between performance and complexity. The selected policies are: simple mapping, with each row of banks making up a bank set; broadcast search; promotion in the adjacent bank upon each hit (one bank per one hit).

A fundamental parameter of the NUCA on-chip network is the latency of wires, that is the latency of transmissions on network links. Firstly, the length of links was determined according to the physical dimensions of banks, which were derived from CACTI. Then, we calculated link latencies applying the delay-optimal repeated wire model proposed by Ho [29]. Since CACTI employs this same model to determine the latency of wiring connections for traditional UCA caches, this approach leads to a general uniformity of the performed analysis. Furthermore, we used the same technological parameters for wires used by CACTI, which in turn are derived from the projections of the ITRS report; from the two proposed projection scenarios, that is aggressive or conservative, we selected the conservative one. We assume that network links (for NUCA) and high-level interconnections (for UCA) are both based on semi-global wires [28], that is wires belonging to an intermediate on-chip metal level, and global wires are reserved for distributing power/ground, clock and critical control signals. After the application of the selected physical model, the transmission of a single flit on a link takes two clock cycles for the best performing S-NUCA configuration and one cycle for the D-NUCA one. The

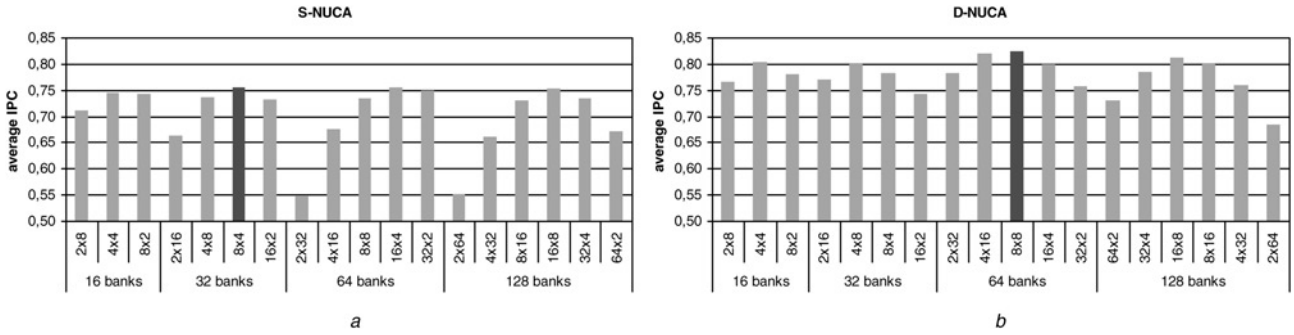

Figure 3 *Performance levels for different bank configurations. The average IPC over the entire workload is assumed as a reference metric*

a S-NUCA architecture
b D-NUCA architecture

design of a NUCA cache is largely influenced by the physical parameters of cache banks, primarily by their height/width. This work focuses on a synchronous NUCA on-chip network, so, in order to achieve high-performance levels, the latency of wires should not be too far from an integer multiplier of the clock cycle time. In fact, the best performing configurations for S-NUCA and D-NUCA obey to this principle.

For the NUCA on-chip network, the link width is 128 bits (16 B) for each direction, as the flit size; this means that a flit can be transmitted on a link once in a row. Regarding the structure of the network packets, we make the following assumptions: the first flit of a packet (header flit) always contains the address of the cache line (physical address, 42 bits) that is involved in the operation, plus some control information for storing the type of the operation and for managing the fragmentation of a packet into flits; four extra flits are necessary only if the operation involves the transfer of the content of the cache line, since each cache line is 64 B wide. For example a request packet for a READ operation requires only one flit, whereas a reply packet for a READ or a request packet for a WRITE operation requires five flits.

The entire analysis is based on the 65 nm technology node. The different architectures considered in this work assume a 16 FO4 clock cycle time, which roughly corresponds to a 5 GHz operating frequency [28].

5 Simulation methodology

The simulation platform adopted for this study is based on an extended version of the cycle-accurate execution-driven simulator sim-alpha [30]. The original version of sim-alpha has been augmented to reproduce the behaviour of a single processor system backed by a UCA, S-NUCA or D-NUCA L2 cache. The level of detail of the NUCA architecture model allows to specify different parameters for the on-chip network, including the latency of transmissions on links (differentiated between vertical and horizontal links, since the aspect ratio of NUCA cache banks may differ from unity), and the buffer capacity and latency of routers. The network model that has been employed in our experiments is cycle accurate and it is able to accurately model the contention on links and buffering resources; at each clock cycle, the model checks the state of the input ports, of the input buffers and of the attached links for each router, updates it and, according to the selected policies described in Section 3 and when no conflicts occur, it triggers the necessary transmission events. The original sim-alpha model for cache banks has been augmented to support a customisable cycle time, that is the minimum interval between two consecutive requests that can be issued: both access time and cycle time can now be specified for cache banks, thus offering a better modelling accuracy, especially when applications exhibit a high L1 miss-rate and/or a high degree of burstiness' of L2 accesses. In this study we assume that UCA caches employ the necessary latching logic to support multiple on-going accesses, spaced out by the proper cycle time.

The simulated systems are based on the Alpha 21264 microprocessor and their configuration parameters are reported in Table 1. Each of the considered systems assumes a single CPU with splitted L1 instruction and data caches, and an on-chip L2 cache whose size is fixed at 8 MB. The architecture of the L2 cache is varied between UCA, S-NUCA and D-NUCA, assuming the configuration parameters reported in Table 1, which were determined as described in Section 4.

Our analysis was performed assuming the workload listed in Table 2, which comprises applications from the SPEC CPU2000 and the NAS Parallel Benchmarks suites. To reduce the overall simulation time, for each application we selected a representative phase of the entire execution, applying the same methodology described by Kim *et al.* [1]. Table 2 reports, for each benchmark, the number of instructions that were skipped from the start (FFWD) and

Table 1 Configuration parameters for the CPU and the memory hierarchy

Parameter	Value
technology node	65 nm
CPU	Alpha 21264
fetch/issue/ commit width	4/4 int. + 2 f.p./11
functional units	4 int. ALUs, 4 int. MUL/DIVs, 1 f.p. ALU, 1 f.p. MUL
Instr. L1 cache	64 kB, 2-way s.a., 64 B line, 1 cycle hit latency
Data L1 cache	64 kB, 2-way s.a., 64 B line, 3 cycles hit latency, 2 ports
L1 caches MSHR size	eight entries, each points up to four targets
main memory latency	300 cycles
UCA L2 cache	8 MB, four-way s.a., 64 B line, access time = 37, cycle time = 4 (cycles)
S-NUCA L2 cache	8 MB, 64 B line, 32 banks (8 × 4), each one four-way s.a. with access time = 13, cycle time = 3 (cycles)
D-NUCA L2 cache	8 MB, 64 B line, 64 banks (8 × 8), each one direct mapped with access time = 11, cycle time = 2 (cycles)

doi: 10.1049/iet-cdt.2008.0078

Table 2 Benchmarks selected for this study, their running phases and their load on the L2 cache in terms of load accesses per million instructions

Application	Suite	FFWD	Run	L2 load accesses/ million instr.
art	SPECFP	2.2 B	200 M	136 500
applu	SPECFP	267 M	650 M	43 300
bt	NPB	800 M	650 M	34 500
bzip2	SPECINT	744 M	1.0 B	9300
cg	NPB	600 M	200 M	113 900
equake	SPECFP	4.459 B	200 M	41 100
galgel	SPECFP	4.0 B	200 M	44 600
gcc	SPECINT	2.367 B	300 M	25 900
mcf	SPECINT	5.0 B	200 M	260 620
mesa	SPECFP	570 M	200 M	2500
mgrid	SPECFP	550 M	1.06 B	21 000
parser	SPECINT	3.709 B	200 M	14 400
perlbmk	SPECINT	5.0 B	200 M	26 500
sp	NPB	2.5 B	200 M	67 200
twolf	SPECINT	511 M	200 M	22 500

the number of simulated instructions (RUN), together with the average load on the L2 cache in terms of load accesses per million instructions.

6 Results

This section reports the results of our analysis. For NUCA caches, we performed a set of simulations varying the cut-through latency from zero to five clock cycles. When the cut-through latency is 0, we assume that the hop latency is given only by the link latency, whose length is equal to the width/height of cache banks; however, all the internal activities of the routers are modelled in detail as for the other cut-through latency values. For each simulation point, we also varied the buffering capacity of routers, selecting three different configurations: five flits per input queue, ten flits per input queue and infinite buffering capacity (ideal case). These values were selected because the size of one of the most frequently occurring packet types on the network is five flits: one header flit, storing control parameters and the cache line address, plus four flits for storing the cache line content (64 B), as described in Section 4. All the collected results are compared against the ones achieved by the L2 UCA architecture, whose configuration has been described in the previous section. In the following, the average IPC over the entire workload is selected as synthetic metric to quantitatively represent the performance level of the systems under evaluation.

Fig. 4 shows the average IPC as the cut-through latency increases from zero to five cycles, for ten flits per channel buffering capacity. We can highlight that the overall system performance is highly sensitive to the cut-through latency. Although D-NUCA always outperforms S-NUCA, the performance of NUCA-based architectures

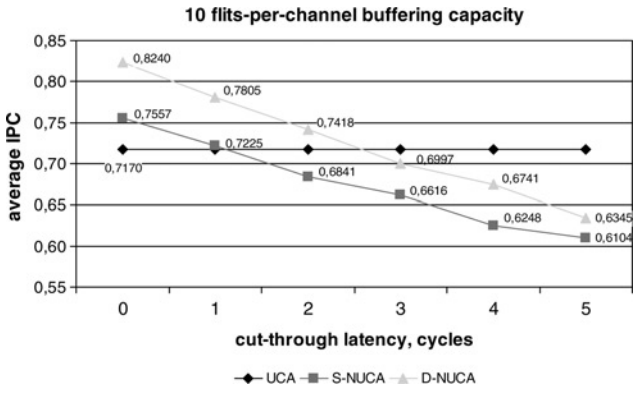

Figure 4 *IPC against cut-through latency. This chart shows how the performance of the overall system changes as the cut-through latency varies from zero to five clock cycles, for both S-NUCA and D-NUCA architectures, when the buffering capacity is fixed at ten flits per channel; the performance achieved by a traditional UCA-based system is reported for comparison*

rapidly decreases from a simulation node to the next. For two cycles latency, the performance level of S-NUCA is lower than UCA, while the benefits of employing a D-NUCA are poor (only 3.5% improvement over UCA). For this node, the extra effort that would be needed to design the communication infrastructure commonly adopted by NUCA architectures would not be acceptable. As expected, for latency values beyond three cycles, things get even worse. However D-NUCA still outperforms S-NUCA. This high sensitivity witnesses that the delay introduced by the on-chip network has strong effects on the overall system performance, according to previous studies [14, 15], while the latency of bank accesses becomes less influential as we move towards higher latencies for routers. However, if routers are able to deliver flits in a single cycle, D-NUCA caches offer a good improvement over UCA (+8.9%). This improvement is larger, as expected, for the null cut-through latency case (+14.9%). These results clearly show that multi-cycle router architectures are not adequate for NUCA; instead, custom architectures tailored to this specific scenario should be used. Even a two cycles routing delay introduces an unacceptable performance degradation.

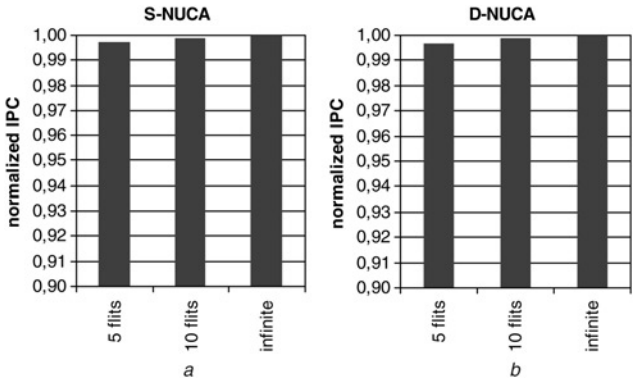

Figure 5 *Performance degradation because of limited buffering capacity. This figure shows the normalised IPC with respect to the ideal case with infinite buffering capacity. These values refer to the simulation node corresponding to one cycle cut-through latency*

a S-NUCA architecture
b D-NUCA architecture

Focusing on a single value for cut-through latency, for example one cycle, it is possible to quantitatively evaluate the performance degradation because of the limited buffer capacity with respect to the ideal router case (infinite buffering capacity), for both S-NUCA and D-NUCA. Fig. 5 shows this degradation, reporting the average IPC over the entire workload, normalised with respect to the ideal router case with infinite buffering capacity. The resulting performance degradation is negligible even for the five flits buffering capacity; for both S-NUCA and D-NUCA the degradation is less than 0.5%. This result suggests that limited buffering capabilities do not jeopardise the performance improvements introduced by NUCA structures.

The limited performance sensitivity to the amount of buffering resources may be explained by analysing the average network traffic, in terms of buffer occupancy. Fig. 6 shows the distribution of the buffer occupancy for two applications: we selected the parser benchmark, which exhibits a moderate load on the network, as witnessed by the utilisation of the link that experiences the highest occupancy (the link is occupied for the 1.9% of the time), and the gcc benchmark, which experiences a relatively higher load (being the link with highest occupancy transmitting for the 15.9% of time). The configuration consists of a D-NUCA architecture, with single-cycle

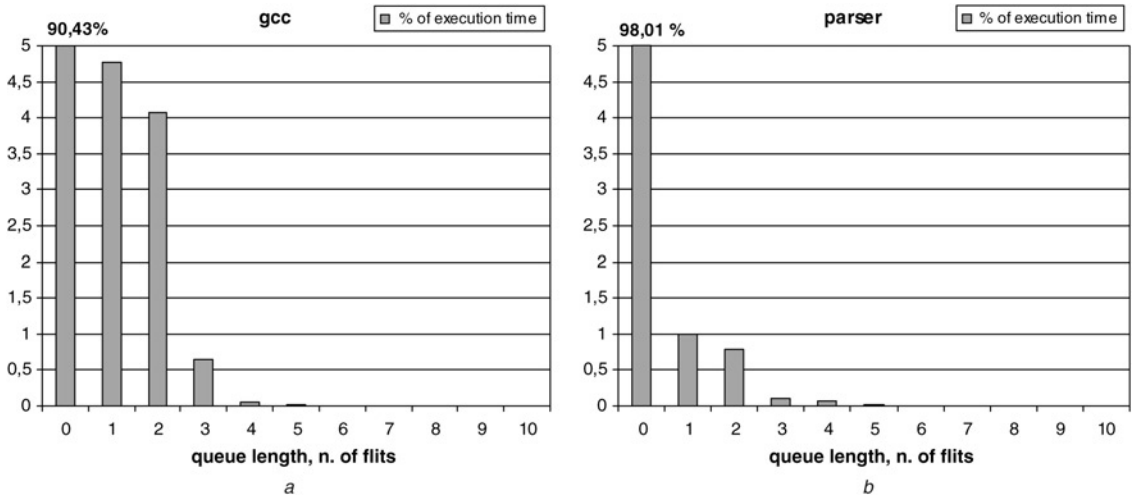

Figure 6 *Distribution of buffer occupancy. The percentage of total execution time spent for each occupancy state is shown. The selected queues belong to the injection point of the NUCA on-chip network. Data refer to a D-NUCA architecture, with single-cycle cut-through latency and infinite buffering capacity*

a Gcc
b Parser

IET Comput. Digit. Tech., 2009, Vol. 3, Iss. 5, pp. 501–512
doi: 10.1049/iet-cdt.2008.0078

cut-through latency and infinite buffering capacity. The queue length distribution is shown for the router located at the injection point of the network, which shows the highest average queue length for all the applications, since this router has to propagate all the traffic generated by the cache controller (the most loaded link is always one of the links connected to this router.). We selected the queue that experiences the highest average occupancy w.r.t. the other queues of the router. For the gcc benchmark, the queue length at the injection point is null (meaning that no buffering resources are occupied) in 90.43% of the time; the maximum measured queue length is 17 flits, but a queue longer than five flits is found with a very low frequency (less than 0.6% of the time), whereas a queue longer than ten flits is found with a frequency lower than 0.001%. The parser benchmark experiences an even lower load condition, being the maximum measured queue length seven flits, but with an occupancy of more than five flits being found only in the 0.002% of the time.

7 Reducing sensitivity to routing latency

The results shown in the previous section indicate that the latency introduced by the on-chip network has a significant impact on the average access latency of a NUCA cache and, as a consequence, on the overall performance of the system. One of the most effective ways to mitigate this effect is to reduce the average number of hops that the cache accesses experience. This can be achieved by reducing the number of cache banks (assuming a constant cache capacity, this means that the size of banks is increased) or clustering the banks so that each cluster is attached to a network node, while keeping the bank size fixed.

The results of applying the first solution can be derived from the analysis that we performed to identify the best bank configurations (Fig. 3): decreasing the number of banks, for both S-NUCA and D-NUCA architectures, leads to lower performance levels. For the other solution, that is the clustered approach, since a wire-delay-dominated environment puts strong constraints on the topology, the only relevant scheme that we took into account for the clustered approach is a configuration with four banks per cluster, as depicted in Fig. 7: an higher number of banks per node would introduce significant wire delays to reach a bank belonging to a cluster from the corresponding router. The partitioning of the address space inside a single cluster is obtained by checking the least significant bits from the index field of the address.

Focusing on D-NUCA architectures, which offer the highest performance level, we observed that modifying the underlying network topology is not enough to significantly reduce the average access latency. In order to achieve a significant improvement, we also introduced an alternative logical organisation, which involves the way with which

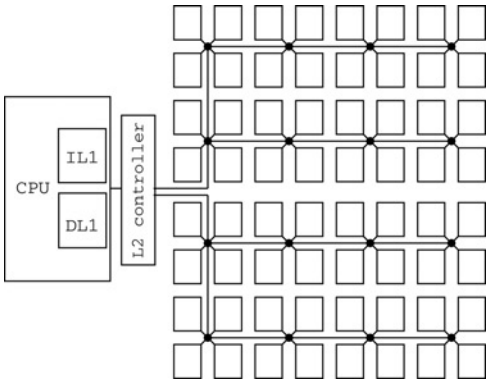

Figure 7 *NUCA on-chip network topology with four banks per node (clustered approach).*

lines are mapped onto cache banks, while in the reference scheme each bank set is mapped onto a single row of banks, with the clustered approach each bank set is mapped onto a row of clusters. Each column of clusters now behaves like a single way of a set associative cache. We performed an analysis similar to the one described in Section 4 to identify the best bank configurations for the clustered approach: since the optimal bank configuration corresponds to the one identified for the D-NUCA reference scheme (64 banks with 128 kB capacity), the clustered D-NUCA with four banks per node globally behaves like a four-way set associative cache. Although the reference scheme assumes a one bank per one hit promotion policy, in the clustered D-NUCA the promotion policy becomes one cluster per one hit.

Fig. 8 reports the performance achieved by the new scheme, when applied to both S-NUCA and D-NUCA architectures. Except for the null cut-through latency case, the clustered scheme always outperforms the reference one; for D-NUCA, the improvement over the UCA scheme is significant: +12.2% at one cycle cut-through latency, and +9.5% at two cycles. These results indicate that the minimal cut-through latency constraint can be relaxed, as this configuration is

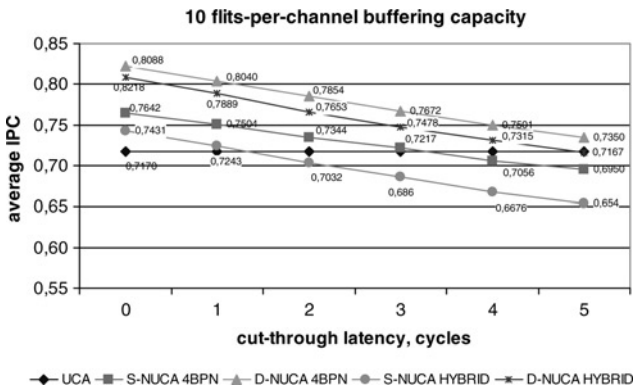

Figure 8 *IPC against cut-through latency for the clustered approach, compared with the hybrid network approach (4BPN = 4 banks per node)*

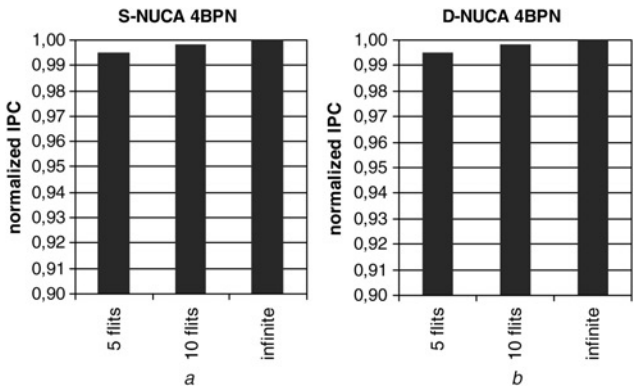

Figure 9 *Performance degradation because of the limited buffering capacity for the clustered approach*

much more scalable w.r.t. the reference architecture. Fig. 9 shows the performance degradation because of the limited buffering capacity for the clustered scheme. The degradation is negligible for both S-NUCA and D-NUCA, analogously to the reference architecture case.

The clustered scheme, while being better performing, reduces the number of network routers, thus leading to a simpler implementation. The additional overhead is given by the additional ports to connect each router to its local banks (three additional ports w.r.t. the traditional scheme). However, a solution based on the multiplexing of a single port to connect to the local banks could be used, employing a simple arbiter. We performed a set of simulations which indicated that the performance degradation because of the loss of parallelism introduced by this solution is negligible, for example for a D-NUCA with a single-cycle cut-through latency and infinite buffering capacity, the performance degradation is only 0.26%.

In the context of NUCA caches, another technique that is able to reduce the number of hops, thus potentially being able to reduce the overall latency of the cache, is the hybrid network approach proposed by Muralimanohar and Balasubramonian [14], which combines point-to-point links to busses. In order to forward request packets, an additional central row of routers is employed and the banks belonging to a column are connected to one of these routers through a shared bus. In order to forward reply packets, a plain 2D mesh is rather used; the reason behind this is that request and reply packets have different bandwidth requirements. Since the 2D mesh is employed for the reply packets, the number of routers of this solution is higher than the number of routers of the clustered approach; for instance, the selected S-NUCA clustered scheme employs eight routers, whereas the selected S-NUCA hybrid approach employs 28 routers. We extended our simulation platform in order to model the hybrid network approach, for both S-NUCA and D-NUCA architectures, and we performed the same set of experiments as for the clustered approach, varying the cut-through latency from zero to five clock cycles. The baseline bank configurations for the hybrid schemes are the same as described in Table 1. Fig. 8 shows the results of this comparison: the hybrid network, because of the reduced number of routers, scales better than a baseline NUCA scheme when the routing latency increases, similarly to the clustered approach. However, the performance levels achieved by the hybrid network approach are always lower than the corresponding ones achieved by the clustered approach, for both S-NUCA and D-NUCA. Mainly, the reason behind this difference derives from geometrical properties of the different NUCA structures: the average latencies calculated statically by assuming a uniform access distribution on the banks and considering only the latency of wires and cache banks (i.e. without taking into account any effect because of network contention and concurrency between different accesses) are lower for the clustered approach. Indeed, for an S-NUCA clustered scheme with null cut-through latency (only wire delay is included) the statically computed latency is 24, whereas for the corresponding hybrid S-NUCA is 29; for one cycle cut-through latency the latencies are 26 and 32 cycles, respectively. These considerations can be applied also to D-NUCA, but in this case we must take into account the distribution of the accesses, because typically it is not uniform due to the migration mechanism. This static analysis can be applied because of the low load condition of the network, as the low sensitivity to the buffering capacity that our analysis suggests. We validated this first insights by analysing the actual average latency values obtained through cycle-accurate simulation, as reported in Fig. 10, the average latencies of the cache roughly track the values obtained through static analysis, thus explaining the difference

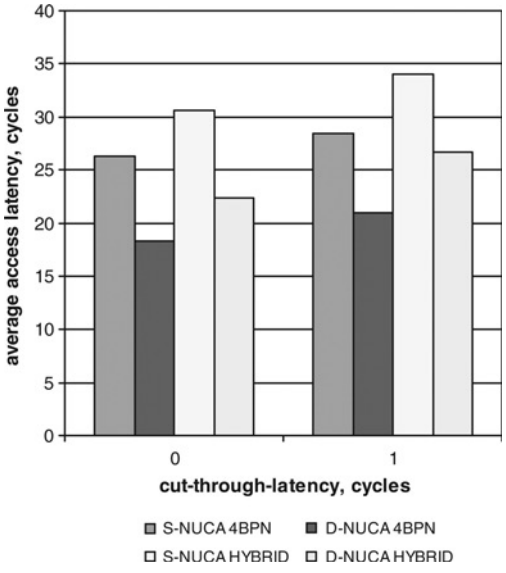

Figure 10 *Average access latency for the clustered approach and for the hybrid network, for null and single cut-through latency (the buffering capacity is fixed at ten flits per channel). These values have been obtained through cycle-accurate simulation*

IET Comput. Digit. Tech., 2009, Vol. 3, Iss. 5, pp. 501–512
doi: 10.1049/iet-cdt.2008.0078

of performances between the clustered and the hybrid approaches.

8 Conclusions

The study described in this paper investigates the impact of two main on-chip network parameters, that is the cut-through latency and the buffering capacity of routers, on the overall performance of uniprocessor systems employing a NUCA L2 cache under realistic workloads. The entire analysis is based on cycle-accurate execution-driven simulation, with a detailed modelling of processor, memory and network behaviour. We have assumed a UCA architecture as a reference performance level, in order to identify under which conditions the NUCA scheme outperforms the traditional UCA. The results indicate that NUCA-based systems exhibit a high sensitivity to the cut-through latency, thus meaning that latency-oriented router architectures (with single-cycle cut-through latency or less) are needed; moreover, varying the buffer capacity has almost negligible effects on the overall performance and the load on the network buffering resources is moderate. From these considerations, we have identified an alternative NUCA organisation that is better performing and is much less sensitive to variations of the router latency. This configuration is based on the clustering of banks, assuming four banks per node, and it is able to relax the strong constraint on latency-oriented routers, thus meaning that multi-cycle routers could be employed in such a scheme with a small performance degradation. We have also compared the performance levels achieved by our technique to the ones achieved by the hybrid network approach, a latency reduction technique introduced by Muralimanohar and Balasubramonian [14], the hybrid approach, while succeeding in reducing this sensitivity, always exhibits lower performance levels.

In a future work we plan to extend the analysis described in this paper to CMP architectures. In order to do this, however, we first need to investigate different aspects of the memory hierarchy design space, such as the balance between private/shared caches, the topology of the NUCA on-chip network (in particular, the position of the cores w.r.t. the cache banks) and the definition of scalable coherence protocols.

9 Acknowledgments

We wish to thank the anonymous reviewers for their helpful and valuable comments. We also wish to thank Stephen Keckler who furnished us with the initial version of the modified sim-alpha simulator, José Duato for his suggestions on our work and Cristian Croce for helping us in the development of the simulation platform.

This work is partially supported by the SARC project funded by the European Union under contract no. 27648.

10 References

[1] KIM C., BURGER D., KECKLER S.W.: 'An adaptive, non-uniform cache structure for wire-delay dominated on-chip caches'. Proc. 10th Int. Conf. Architectural Support for Programming Languages and Operating Systems, San Jose, CA, USA, October 2002, pp. 211–222

[2] AGARWAL V., HRISHIKESH M.S., KECKLER S.W., BURGER D.: 'Clock rate versus IPC: the end of the road for conventional microarchitectures'. Proc. 27th Int. Symp. Computer Architecture, Vancouver, Canada, June 2000, pp. 248–259

[3] BENINI L., DE MICHELI G.: 'Networks on chips: a new SoC paradigm', *IEEE Comput.*, 2002, **35**, (1), pp. 70–78

[4] DALLY W.J., TOWLES B.: 'Route packets, not wires: on-chip interconnection networks'. Proc. 38th Design Automation Conf., Las Vegas, NV, USA, June 2001, pp. 684–689

[5] DALLY W.J., TOWLES B.: 'Principles and practices of interconnection networks' (Morgan Kaufmann, 2003)

[6] KROFT D.: 'Lockup-free instruction fetch/prefetch cache organization'. Proc. 8th Int. Symp. Computer Architecture, Minneapolis, MN, USA, May 1981, pp. 81–87

[7] BARDINE A., FOGLIA P., GABRIELLI G., PRETE C.A., STENSTRÖM P.: 'Improving power efficiency of D-NUCA caches', *ACM SIGARCH Comput. Arch. News*, 2007, **35**, (4), pp. 53–58

[8] FOGLIA P., MANGANO D., PRETE C.A.: 'A cache design for high performance embedded systems', *J. Embedded Comput.*, 2005, **1**, (4), pp. 587–597

[9] CHISHTI Z., POWELL M.D., VIJAYKUMAR T.N.: 'Distance associativity for high-performance energy-efficient non-uniform cache architectures'. Proc. 36th Int. Symp. Microarchitecture, San Diego, CA, USA, December 2003, pp. 55–66

[10] HUH J., KIM C., SHAFI H., *ET AL.*: 'A NUCA substrate for flexible CMP cache sharing'. Proc. 19th Int. Conf. Supercomputing, Cambridge, MA, USA, June 2005, pp. 31–40

[11] BECKMANN B.M., WOOD D.A.: 'Managing wire delay in large chip-multiprocessor caches'. Proc. 37th Int. Symp. Microarchitecture, Portland, OR, USA, December 2004, pp. 319–330

[12] CHISHTI Z., POWELL M.D., VIJAYKUMAR T.N.: 'Optimizing replication, communication, and capacity allocation in CMPs'. Proc. 32nd Int. Symp. Computer Architecture, Madison, WI, USA, June 2005, pp. 357–368

[13] MURALIMANOHAR N., BALASUBRAMONIAN R., JOUPPI N.P.: 'Optimizing NUCA organizations and wiring alternatives for large caches with CACTI 6.0'. Proc. 40th Int. Symp. Microarchitecture, Chicago, IL, USA, December 2007, pp. 3–14

[14] MURALIMANOHAR N., BALASUBRAMONIAN R.: 'Interconnect design considerations for large NUCA caches'. Proc. 34th Int. Symp. Computer Architecture, San Diego, CA, USA, June 2007, pp. 369–380

[15] JIN Y., KIN E.J., YUM K.H.: 'A domain-specific on-chip network design for large scale cache systems'. Proc. 13th Int. Symp. High-Performance Computer Architecture, Phoenix, AZ, USA, February 2007, pp. 318–327

[16] TAYLOR M.B. *ET AL.*: 'The Raw microprocessor: a computational fabric for software circuits and general purpose programs', *IEEE Micro*, 2002, **22**, (2), pp. 25–35

[17] WENTZLAFF D. *ET AL.*: 'On-chip interconnection architecture of the Tile processor', *IEEE Micro*, 2007, **27**, (5), pp. 15–31

[18] VANGAL S. *ET AL.*: 'An 80-tile 1.28TFLOPS network-on-chip in 65 nm CMOS'. Digest of Technical Papers, Int. Solid-State Circuits Conf., San Francisco, CA, USA, February 2007, pp. 98–589

[19] SANKARALINGAM K. *ET AL.*: 'Distributed microarchitectural protocols in the TRIPS prototype processor'. Proc. 39th Int. Symp. Microarchitecture, Orlando, FL, USA, December 2006, pp. 480–491

[20] SANKARALINGAM K., SINGH V.A., KECKLER S.W., BURGER D.: 'Routed inter-ALU networks for ILP scalability and performance'. Proc. 21st Int. Conf. Computer Design, San Jose, CA, USA, October 2003, pp. 170–177

[21] BALFOUR J., DALLY W.J.: 'Design tradeoffs for tiled CMP on-chip networks'. Proc. 20th Int. Conf. Supercomputing, Queensland, Australia, June 2006, pp. 187–198

[22] PEH L.-S., DALLY W.J.: 'A delay model and speculative architecture for pipelined routers'. Proc. 7th Int. Symp. High-Performance Computer Architecture, Nuevo Leone, Mexico, January 2001, pp. 255–266

[23] MULLINS R., WEST A., MOORE S.: 'Low-latency virtual-channel routers for on-chip networks'. Proc. 31st Int. Symp. Computer Architecture, München, Germany, June 2004, pp. 188–197

[24] WANG H., PEH L.-S., MALIK S.: 'Power-driven design of router microarchitectures in on-chip networks'. Proc. 36th Int. Symp. Microarchitecture, San Diego, CA, USA, December 2003, pp. 105–116

[25] WANG H., PEH L.-S., MALIK S.: 'A power model for routers: modeling Alpha 21364 and InfiniBand routers', *IEEE Micro*, 2003, **23**, (1), pp. 26–35

[26] WILTON S.J., JOUPPI N.P.: 'CACTI: an enhanced cache access and cycle time model', *IEEE J. Solid-State Circuits*, 1996, **31**, (5), pp. 677–688

[27] THOZIYOOR S., MURALIMANOHAR N., JOUPPI N.P.: 'CACTI 5.0'. HP Technical Report (HPL-2007-167), 2007

[28] International Technology Roadmap for Semiconductors: Edition Report, 2005

[29] HO R.: 'On-chip wires: scaling and efficiency'. PhD thesis, Stanford University, 2003

[30] DESIKAN R., BURGER D., KECKLER S.W., AUSTIN T.: 'Sim-alpha: a validated, execution-driven Alpha 21264 simulator'. Technical Report (TR-01-23), Department of Computer Sciences, University of Texas at Austin, 2001

doi: 10.1049/iet-cdt.2008.0078