Adaptive Routing and Dynamic Frequency Scaling for NoC Power-Performance Optimizations

Davide Zoni
Politecnico di Milano - DEIB
20133 Milano, ITALY
Email: davide.zoni@polimi.it

José Flich
Universitat Politècnica de València - GAP
Valencia, SPAIN
Email: jflich@disca.upv.es

William Fornaciari
Politecnico di Milano - DEIB
20133 Milano, ITALY
Email: william.fornaciari@polimi.it

***Abstract*— On-chip networks (NoCs) promise to become an efficient communication infrastructure for multi-core architectures. However, there is still a need for efficient power-performance methodologies, since the interconnect power-envelope is really slim and cannot be neglected. Indeed, new power-aware design explorations in current and future multicore systems are needed.**

The possibility to use different router microarchitectural options and different routing algorithms to increase performance, combined with standard power-aware mechanisms, i.e. DVFS and Power Gating techniques, provides a huge design space to be explored. In this perspective, this paper presents a comparative analysis of different NoC routing algorithms combined with Dynamic Frequency Scaling (DFS).

I. Introduction

Networks-on-Chip (NoC) represent a flexible solution to manage the communication in multi-core and many-core. However, the interconnection may consume a significant part of the total chip power, thus can not be neglected [1]. Moreover, the advances in semiconductor technology scaling allow to create larger and more powerful circuits, fueling the growth of multi-core chip. However, the actual integration density is limited by the reliability of the device. For example, actual digital designs are more susceptible to sources of parametric variations producing unexpected power/performance fluctuations, which are becoming a major fabrication challenge for the upcoming technology scales [2]. Moreover, aging mechanisms, i.e. Negative Bias Thermal Instability, represent additional sources of faults, that must be accounted for during the design. To this extent, there is a need for methodologies allowing to steer the power-performance trade-off accounting for process variation (PV) and aging issues as well. In this perspective, multiple solutions have been proposed focusing on different areas, i.e. microarchitectural solutions, routing algorithms or Dynamic Voltage and Frequency Scaling (DVFS). For example, adaptive routing [3] allows to flexible manage traffic, reducing NoC latency and faces both PV and aging at runtime modifying the routes. On the other hand, the Global Asynchronous Local Synchronous (GALS) design paradigm [4] allows to partition the chip in a number of frequency islands that are synchronous on the inside, and asynchronous at the boundaries. Thus, each island has the required amount of computational power and the power metric is optimized.

Although many methodologies have been presented, few of them focus on a combination of different basic solutions to even sharp the power-performance trade-off. In this perspective, this paper presents a comparative analysis focused on routing algorithms and DFS and GALS schemes to optimize the power-performance figure of merit in two different steps. First, we compared different adaptive routing algorithms against deterministic routing focusing on the power-performance trade-off we get when the NoC router frequencies are heterogeneous due to both GALS design or because PV. Second, we explored the combination of routing algorithms with DFS mechanisms to further improve power and performance. In this perspective the paper shows how cross-layers techniques can be used jointly to improve the performance of the multi-core.

All the DFS policies and the routing algorithms have been validated using a cycle accurate simulator that has been enhanced to support adaptive routing as well as GALS NoC and an accurate PLL model to mimic the dynamic frequency scaling feature.

The rest of the paper is organized in three parts. Section II presents the state of the art related to both adaptive routing and the dynamic frequency scaling techniques for power-performance optimization. The modified simulator flow including the adaptive routing support as well as the GALS NoC and the DFS module is presented in Section III, while results are reported in Section IV. Last, conclusions are devoted in Section V.

II. Related works

While a lot of works address the power-performance trade-off none of them exploit at the same time frequency and adaptive routing to improve the power-performance trade-off.

In [5], authors presents a fine-grained frequency tuning scheme for NoC router to optimally manage the power-performance trade-off, using a deterministic *XY* routing algorithm. The work in [6] leverages the traffic unbalancing within a specific NoC topology to exploit the classical technique of DVFS to minimize the power consumption coupled with ad-hoc routing algorithms. However, the solution relies on a mathematical formulation that must be solved at design time

The work has been partially funded by the HiPEAC Collaboration Grant *"Dynamic router model and control to optimize power-performance tradeoff in NoCs"*.

978-1-4799-1170-7/13/$31.00 ©2013 IEEE

considering a static even in average behavior for the system as a whole. Thus, it is not possible to face run-time variations in network load and failures. The work in [7] proposes an heuristic approach focused on DVFS actuators to mitigate power consumption on the real Intel SCC multi-core. Even if this solution has been tested on a real multi-core, it does not provide an accurate model of the relation between frequency and performance thus it does not allow to exploit the solution for further improvements. In [4] and [8], authors provide a design-time frequency island partitioning scheme. Moreover, [8] coupled the frequency island partitioning process with an ad-hoc routing algorithm. However, such proposals can not adapt to run-time failure or traffic variations. The work in [9] presents an adaptive routing scheme to compensate for performance heterogeneity due to on-die variations or traffic imbalance. Another adaptive routing scheme is discussed in [10] with focus on global adaptive routing to overcome the suboptimal local adaptive ones. However both [9] and [10] do not consider any impact on routing due to policies acting on frequency.

III. Proposed evaluation

This section discusses the baseline adaptive routing scheme as well as the Global Asynchronous Local Synchronous (GALS) design methodology and the Dynamic Frequency Scaling (DFS) module integrated in the simulation framework.

We start from the GEM5 simulator [11], that is an event-driven simulation framework for multi-core architectures enabling an accurate NoC model [12]. The baseline simulator has been modified introducing four different main logical blocks [13].

First, the event management system of the simulator has been extended to support the possibility to move already scheduled events between different simulation times, a feature required to implement the DFS behavior. In particular, we needed a framework that is able to group all the events scheduled for a single component and to move them forward or backwards with respect to the actual scheduled time. Changing the frequency of a component entails moving already scheduled events to the time they will need to be serviced considering the frequency change, and storing the new frequency value, so that subsequently generated events are scheduled at the appropriate time. Moreover, we reorganized the router pipeline model in the Garnet network allowing to move events without introducing semantic misbehavior, as described in the Wattch paper [14], i.e. the stages in a pipeline model for an event driven simulator must be executed in backward order, thus from the last pipeline stage.

Second, we implemented a 2-way handshaking protocol that only adds two single bit lines, i.e. request and acknowledge, to a network link. The resynchronizer enables the communication between different frequency islands. It is worth noticing that our implementation allows to specify flexible frequency islands from a per router granularity to a single global island for the entire multi-core. Thus, in order to simulate an asynchronous multi-core an adequate number of resynchronizer components must be inserted on each frequency domain border, to ensure reliable data transfer as well as to avoid metastability. It is worth noticing we have to insert resynchronizers between each router and the other connected logic, i.e. L1s, L2s and memory controllers, if any, since we assume asynchronous NoC communication between different domains.

TABLE I: Experimental setup: processor and router micro-architectures and technology parameters.

Processor core	1GHz, out-of-order Alpha core
Int-ALU	4 integer ALU functional units
Int-Mult/Div	4 integer multiply/divide functional units
FP-Mult/Div	4 floating-point multiply/divide functional units
L1 cache	64kB 2-way set assoc. split I/D, 2 cycles latency
L2 cache	512kB per bank, 8-way associative
Coherence Prot.	MESI
Router	3-stage wormhole virtual channelled switched with 32bit link width 4fl/VC 2/4 virtual channels (VCs) for each virtual network 3 virtual networks (Garnet network [12])
Topology	2D-mesh 16 core
Technology	45nm at 1.0V

Third, we augmented the Orion [15] power module in the simulator allowing for run-time power estimation even when router frequencies are dynamically changed.

Last, the framework allows for both deterministic and minimal adaptive routing support. In particular, the implemented adaptive routing module partitions the available Virtual Channels (VCs) between adaptive VCs and escape path VCs [3]. The partition is due for all the available virtual networks (VNET). The implemented adaptive routing scheme exploits the work in [3], which demonstrates the possibility to build adaptive routing algorithms, even with cycles in the Channel Dependency Graph (CDG), as an escape deadlock-free path is always available. In particular, we couple the adaptive routing with a deterministic *XY* routing, that is able to use a subset of the available VCs, i.e. escape path VCs, that are virtually separated by the adaptive ones. At each route computation, the routing unit tries to bind an output port with a free adaptive VC to the routed packet. If during the routing stage there are no adaptive VCs on all the minimal adaptive paths, the packet is routed using the escape paths, thus using the deterministic routing and the escape VCs. It is worth noticing that once a packet is routed through an escape path all the subsequent routes until destination must be deterministic.

IV. Results

This section discusses simulation results obtained on a representative 2D-mesh 16-core architecture, where Table I reports detailed parameters we used for the experiments. Each tile is composed of an Alpha 21264 core running at 1GHz, with private L1 cache and a shared L2 cache composed of 16 physically distributed banks, one connected to each router. We simulated different scenarios considering several applications taken from the *MiBench* suite using the simulation toolchain detailed in Section III.

We assess different routing algorithms combined with static frequency islands and dynamic frequency scaling policies.

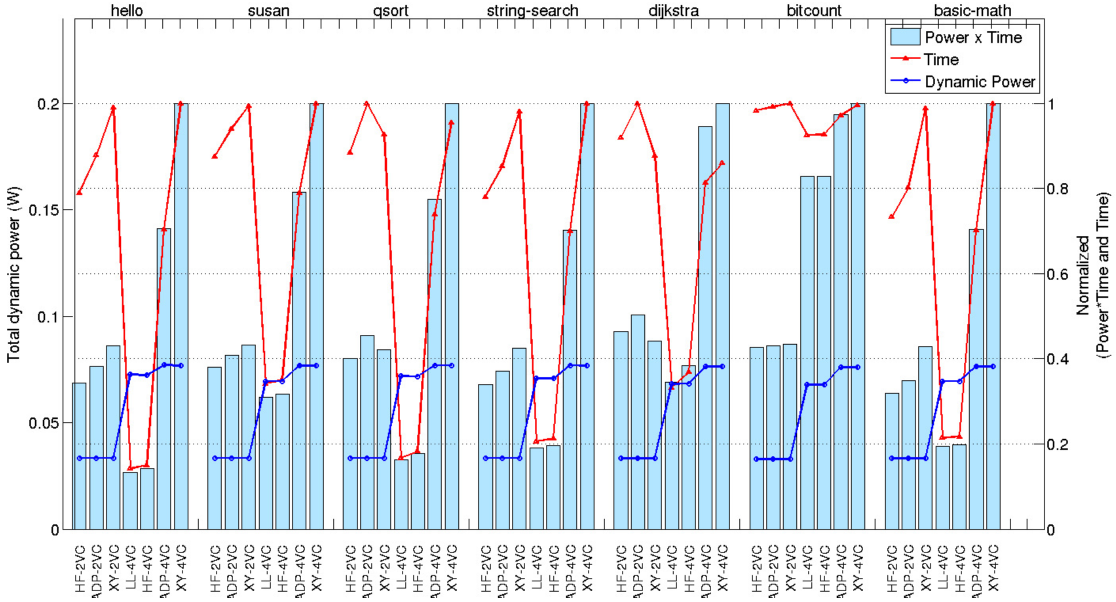

Fig. 1: Results with static frequency islands where the internal routers of the 2D-mesh are running at 100MHz while all the others are running at 1GHz. Note that *LL-2VCs* and *ADP-2VCs* share the same behavior with 2 VCs and a single escape VC.

TABLE II: Evaluated policies organized by classes.

Class	NameID	Details
Deterministic XY	**XY-2VCs**	Classical deterministic routing
	XY-4VCs	
Adaptive Simple **(Minimal)**	**ADP-2VCs**	Looks for an available port on the minimal path, otherwise the packet is switched to the escape path
	ADP-4VCs	
Adaptive High Freq **(Minimal)**	**HF-2VCs**	Looks for an available port on the minimal path ordered by high frequency, otherwise the packet is switched to the escape path
	HF-4VCs	
Adaptive Less Load **(Minimal)**	**LL-2VCs**	Looks for an available port on the minimal path ordered by less loaded output port, otherwise the packet is switched to the escape path
	LL-4VCs	

Table II reports the details related to the routing protocols evaluated in the rest of this section. In particular, results are reported in Figure 1 and Figure 2 using static frequency islands and the DFS module, respectively. Both figures share the same format. The evaluated benchmarks are reported on the top x axis, while the bottom x-axis reports the identifier for the routing algorithm. For each experiment, we reported the time to completion, i.e. red line, the total power averaged on all routers using a blue line and the product of the two metrics, i.e. the blue bar. It is worth to note we normalized both the time to completion and the product of the metrics to better format the figures.

Figure 1 allows for three considerations. First, the adaptive frequency-aware routing, i.e. *HF-2VCs*, obtains a better time-power product in general, since it can exploit the information on the frequency to steer packets. However, the *XY-2VCs* provides a lower time to completion running *dijkstra*. This is due to the low flexibility the adaptive routing algorithm has since only two VCs are available and one of them is used for the escape path. Thus, a packet routed on the escape path can be routed until destination using only a single VC. On the other side, the *XY-2VCs* can always use two VCs despite it is deterministic. Second, the adaptive routing always behaves better than the deterministic one. In particular, the *LL-4VCs* gets a lower power-time product for all the evaluated benchmarks even if comparable with the *HF-4VCs*. This is due to the possibility to collect frequency information indirectly looking for the less used output port at run-time, that in general is the one connected to the higher frequency router. To this extent, *LL-4VCs* is better than *HF-4VCs*, since it can use the already available credit based flow control information without requiring for neighbor information exchange on frequency. Moreover, the *LL-4VCs* has a better power-performance product than all other routing algorithms using 2 VCs. Moreover, *LL-4VCs* and *HL-4VCs* are the only ones that benefit from 4 VCs, showing better results than the ones achieved with 2 VCs, assessing for an effective resource usage, i.e. VCs. Last, despite the different routing algorithms, i.e. deterministic and adaptive, the power consumption without consider the increase due to the different routing policies is mainly dependent on the number of VCs, since all the routing strategies are minimal.

Figure 2 collects results considering a DFS module and a threshold-based policy coupled with different routing algorithms. While the adaptive routing still allows for better performance the use of dynamic frequency scaling module impacts the behavior of the routing. Considering 2 VCs most of the time both adaptive and deterministic routing algorithms share common performance, while the deterministic routing even overcomes the adaptive strategies using *dijkstra*. This unexpected behavior is partially due to the limited adaptive routing scheme flexibility, i.e. one VC is dedicated to the

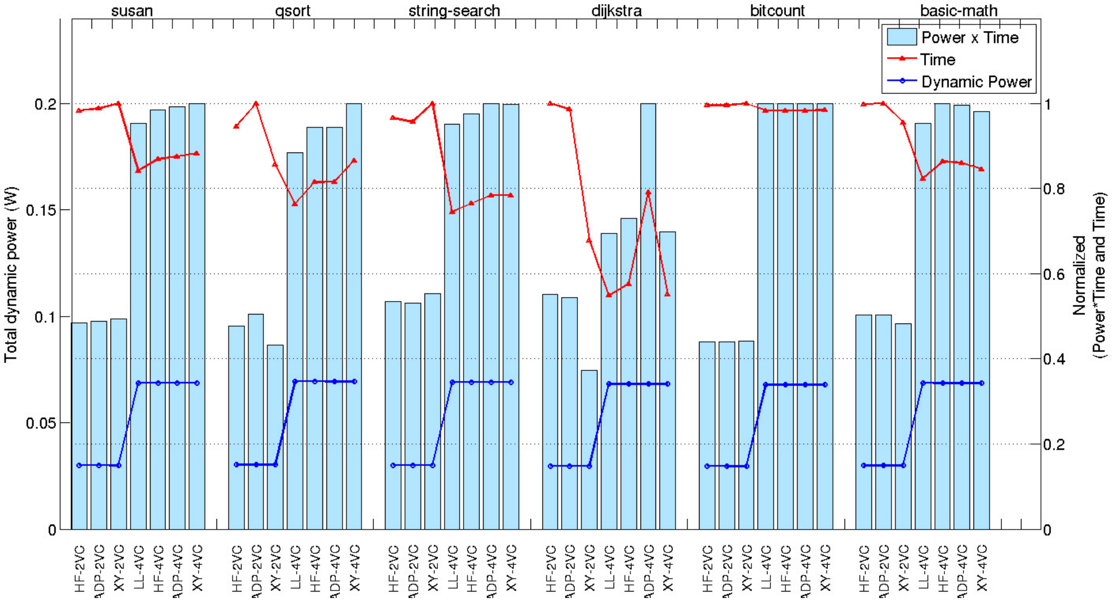

Fig. 2: Results with the threshold policy running on each router. The threshold policy can switch between three frequencies, i.e. 250MHz, 500MHz and 800MHz, depending on two threshold on the buffer filling level of the router, i.e. 10 and 20 flits. Note that *LL-2VCs* and *ADP-2VCs* share the same behavior with 2 VCs and a single escape VC.

escape path. Moreover, the threshold policy senses the contention on the router buffers to manage the frequency. Thus, even if the deterministic routing always uses the same route for each source-destination pair making the route congested, the threshold policy increases the frequency to contrast the issue. Results change using 4VCs, since the *LL-4VCs* always overcomes all the other routing strategies. However, it cannot overcome the power-performance of the 2 VCs routing algorithms, thus highlighting the impact of the DFS.

V. Conclusions

The joint use of dynamic routing and dynamic frequency scaling for NoC-based multi-core can improve the power-performance trade-off. Our experiments show a great impact the number of the VCs has on the routing strategy. In particular, it seems reasonable to use an adaptive routing scheme when the number of VCs is high, i.e. 4, and without a coupled adaptive frequency policy. Second, there is no need to probe frequency information on the neighbor routers to improve the routing both in static, i.e. GALS, and dynamic frequency assignment, i.e. using DFS. To this extent, a simple *LL-VCs* scheme can boost the NoC. Although, the use of a DFS policy coupled with a deterministic routing is better when simple designs and low power are targeted.

References

[1] S. Borkar, “Networks for multi-core chips: a contrarian view,” in *Special Session at ISLPED*, 2007.

[2] C. Nicopoulos, S. Srinivasan, A. Yanamandra, D. Park, V. Narayanan, C. Das, and M. Irwin, “On the effects of process variation in network-on-chip architectures,” *Dependable and Secure Computing, IEEE Transactions on*, 2010.

[3] J. Duato, “A new theory of deadlock-free adaptive routing in wormhole networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 4, no. 12, pp. 1320–1331, Dec. 1993.

[4] U. Ogras, R. Marculescu, P. Choudhary, and D. Marculescu, “Voltage-frequency island partitioning for gals-based networks-on-chip,” in *DAC. 44th ACM/IEEE*, 2007, pp. 110–115.

[5] A. K. Mishra, A. Yanamandra, R. Das, S. Eachempati, R. Iyer, N. Vijaykrishnan, and C. R. Das, “Raft: A router architecture with frequency tuning for on-chip networks,” *Journal of Parallel and Distributed Computing*, vol. 71, no. 5, pp. 625 – 640, 2011.

[6] A. Bianco, P. Giaccone, and N. Li, “Exploiting dynamic voltage and frequency scaling in networks on chip,” in *High Performance Switching and Routing (HPSR), 2012 IEEE*, 2012, pp. 229–234.

[7] R. David, P. Bogdan, R. Marculescu, and U. Ogras, “Dynamic power management of voltage-frequency island partitioned networks-on-chip using intel’s single-chip cloud computer,” in *Networks on Chip (NoCS), 2011 Fifth IEEE/ACM International Symposium on*, 2011, pp. 257–258.

[8] W. Jang, D. Ding, and D. Pan, “A voltage-frequency island aware energy optimization framework for networks-on-chip,” in *ICCAD 2008. IEEE/ACM International Conference on*.

[9] Y. Markovsky, Y. Patel, and J. Wawrzynek, “Using adaptive routing to compensate for performance heterogeneity,” in *NoCS 2009. 3rd ACM/IEEE International Symposium on*.

[10] L. Tedesco, F. Clermidy, and F. Moraes, “A monitoring and adaptive routing mechanism for qos traffic on mesh noc architectures,” ser. CODES+ISSS. New York, NY, USA: ACM, 2009, pp. 109–118.

[11] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, “The gem5 simulator,” *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, Aug. 2011.

[12] N. Agarwal, T. Krishna, L.-S. Peh, and N. Jha, “Garnet: A detailed on-chip network model inside a full-system simulator,” in *ISPASS*, 2009.

[13] F. Terraneo, D. Zoni, and W. Fornaciari, “A cycle accurate simulation framework for asynchronous noc design,” in *SOC 2013, International Symposium on System-on-Chip, Tampere, Finland*, 2013.

[14] D. Brooks, V. Tiwari, and M. Martonosi, “Wattch: a framework for architectural-level power analysis and optimizations,” in *ISCA*. New York, NY, USA: ACM, 2000, pp. 83–94.

[15] A. Kahng, B. Li, L.-S. Peh, and K. Samadi, “Orion 2.0: A fast and accurate noc power and area model for early-stage design space exploration,” in *DATE ’09.*, 2009, pp. 423 –428.