An analytical, dynamic, power-performance router model for run-time NoC optimizations

Davide Zoni, Federico Terraneo and William Fornaciari
Dipartimento di Elettronica, Informazione e Bioingegneria - DEIB
Politecnico di Milano, 20133 Milano, ITALY
Email: {davide.zoni,federico.terraneo,william.fornacia}@polimi.it

Abstract— **Network-on-Chip (NoC) are considered the prominent interconnection solution for current and future many-core architectures. While power is a key concern to deal with during architectural design, power-performance trade-off exploitation requires suitable analytical models to highlight the relations between actuators and such optimization metrics. This paper presents a model of the dynamic relation between the frequency of a NoC router and its performance, to be used for the design of run-time Dynamic Voltage and Frequency Scaling (DVFS) schemes capable of optimizing the power consumption of a NoC. The model has been obtained starting from both physical considerations on the NoC routers and identification from traffic data collected using a cycle-accurate simulator. Experimental results show that the obtained model can explain the dependence of a router congestion on its operating frequency allowing to use it as a starting point to develop power-performance optimal control policies.**

I. Introduction

Many-core architectures are regarded as one of the most promising solution to deliver increasing processing capabilities to future general purpose computing systems.

However, the shift from multi-core to many-core chips will impose even more stringent requirements on the interconnection architecture to provide an increasing number of cores with instruction and data streams to process. This is causing the shift from bus-based interconnection schemes to Network-on-Chip (NoC) based ones.

Another increasingly concerning design challenge is power optimization. As power dissipation in multi-core and many-core systems is no longer almost entirely concentrated in the cores themselves, the contribution of other on-chip components can no longer be considered negligible. In particular, Network-on-Chip can consume a significant part of the total chip power [1]. Therefore, techniques that can reduce the power consumption of NoCs, and in particular explore their power-performance trade-off are of particular interest.

One promising paradigm for NoC design is the Global Asynchronos Local Synchronous (GALS) one [2]. Its usefulness stems from the fact that, as the number of transistors per chip keeps increasing, so does the power consumption required to distribute a skew-free clock throughout the chip. The GALS design paradigm partitions the chip in a number of frequency islands that are synchronous on the inside, and asynchronous at the boundaries. This design methodology can adapted to NoC quite easily, by designing tiles composed of cores, caches and NoC routers, and using asynchronous network links to provide interconnection among tiles [3].

As the design of GALS NoCs requires to introduce resynchronizer circuits or asynchronous First-In First-Out queues (FIFOs) at the tile boundaries to prevent metastability [4], this allows tochange the frequency of tiles independently, and therefore employ Dynamic Frequency Scaling (DFS) and Dynamic Voltage and Frequency Scaling (DVFS) schemes to further optimize the power-performance trade-off.

This work starts from the scenario just sketched, having as aim to develop dynamic models of the performance of a NoC router subject to frequency changes, to allow the development of more refined control schemes with respect to the present state of the art.

A. Novel Contribution

We developed an accurate and flexible dynamic frequency-performance model for the NoC routers to be used for power-performance run-time optimization exploiting frequency scaling as an actuator. In particular the model allows to easily abstract the optimization policy design focusing on the required solution properties. More schematically, the proposal encompasses three different contributions:

- *Frequency to performance dependence* - The presented model explains the relation between the frequency of a router and its performance level. This aspect is of paramount importance to abstract the model and has been verified on an extensive set of experiments.
- *Fine-grained run-time optimization enabler* - The proposed model accounts for the dynamic behavior of the router's performance when a frequency change occurs. To this extent it is possible to evaluate run-time optimization policies that account for the behavior we observed between the time the frequency is changed and the time the level of performance really increases.
- *Flexible and extensible evaluation framework* - The model has been evaluated considering an ad-hoc framework that integrates a GALS model for the NoC developed inside a multi-core simulator. The integrated flow represents a great improvement to the state of the art, since it combines the semantic of a cycle accurate simulation with a complete asynchronous NoC model allowing for DFS power-performance optimizations. Such a framework also accounts for resynchronization overhead

978-1-4799-1166-0/13/$31.00 ©2013 IEEE

at the tile boundaries, by modeling a two way handshake resynchronization scheme similar to the work in [5].

B. Paper Structure

The rest of the paper is organized in three sections. Section II is an overview of the state-of-the-art on NoC models focused on power-performance optimization exploiting frequency scaling actuators as well as early stage simulation toolchain allowing to assess such methodologies. Section III presents the power-performance model. Results are then reported and discussed in Section IV, while conclusions are drawn in Section V.

II. Related works

This sections presents the state of the art on Network-on-Chip power-performance trade-off, focusing on two different aspects related to the DVFS mechanism. First, an overview of different power-performance optimization methodologies will be discussed with particular emphasis on modeling. Then, some frameworks will be presented that allow for power-performance trade-off in the on-chip networks, since they are required to support both NoC modeling as well as NoC power-performance optimization.

[6] discussed a fine-grained frequency tuning scheme for NoC router to optimally manage the power-performance trade-off. In particular the methodology exploits signaling between routers to collect critical information to steer frequency. Moreover, the work allows for a run-time VC reconfiguration to aggressively save power. However, the proposed solution does not model the relations between routers' frequencies and real performance and power measures. Thus the proposed solution cannot be easily improved, since it represents a fixed heuristic.

The work in [7] 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. A power minimization linear programming model has been proposed to find a routing that minimizes the power consumption while satisfying the traffic demands and meeting the link capacity constraint. The solutions relies on a mathematical formulation that must be solved at design time considering an average behavior for the whole system. In contrast, our paper presents a model of the relation between frequency and performance that allows to design run-time optimization policies. Moreover, our solution can implicitly account for run-time variations in requests and failures.

Different proposals focus on the queuing theoretical framework to model on-chip networks. [8] presented analytical model that focuses on QoS assurance. However, it assumes that the NoC has an underlying synchronous behavior with constant service time routers, thus it is not suitable for optimization using the well known DVFS-based actuators. Moreover, it assumes infinite buffers, and taking into account the finite nature of NoC buffers would greatly complicate the model. An analytical queuing theoretical approach to model NoCs accounting for finite buffers has been presented in [9]. In particular the solution exploits the classic queuing theory, while the router serving time model has been derived from real data. However, the methodology relies on exponential distribution for flit arriving times that cannot in general be guaranteed in NoCs [8], and does not account for run-time frequency variations.

The work in [10] 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 order to support such reviewed proposals, there are several simulation frameworks in literature focusing of power-performance exploration. However, few of them are specifically focused to support power-performance trade-off analysis in multi-core scenarios enabling DVFS support for NoC routers. The *Polaris* framework [11] allows for power and area design space exploration for Network-on-Chip architectures without considering a detailed power estimation for both processors and memory hierarchy. Moreover, it does not implement an heterogeneous NoC model to allow for run-time frequency changes during simulation. The work in [12] presents *Sniper*, a framework that can simulate multi-cores underpinned by an on-chip network interconnect. Moreover, the simulator supports per core DVFS. However such support is not present for the NoC model and there is no possibility to consider power-performance trade-off for NoC-based multi-core since the Sniper integration with the McPAT power model [13] is not completed yet. The HANDS [14] framework sits on GEM5 and allows to simulate multi-core architectures collecting power-performance thermal and reliability estimates at the same time. Even if this is quite accurate it lacks a complete asynchronous NoC model, thus it is not possible to test different DVFS schemes to trade-off power vs. performance.

III. Proposed estimation flow

This section presents a model relating the performance of a NoC router to its operating frequency, in the form of a linear, time-invariant dynamic system, estimated from simulations performed on real (i.e., non synthetic) traffic flow.

A. Proposed model

The proposed model is mainly intended to be used for designing fine-grained power-performance optimization schemes that work at the granularity of a single router, by using DFS to change the router frequency at run-time. The frequency to congestion relation was devised starting from a set of physical considerations to derive the model structure, complemented by parameter identification. As a result, the model is of the gray-box type, with its structure and some of its parameters dictated by the underlying process, and the remaining parameters identified based on experimental data. First, we define the inputs and outputs of the model, then we bind them to the concepts of power and performance for a router. Last, we detail the internal structure of the router model, i.e. the state equations. It is worth noticing we consider a virtual channeled

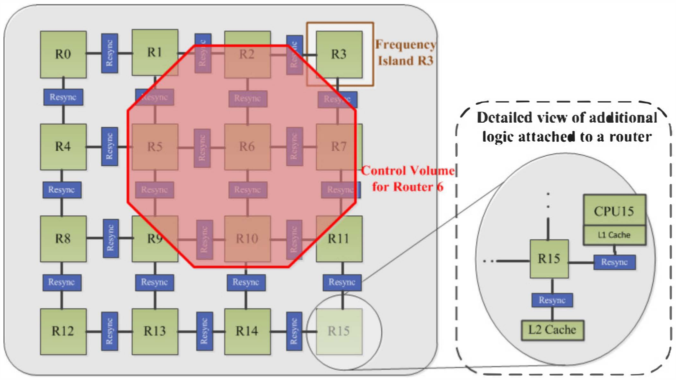


Fig. 1: Control volume for Router 6, in the case when each router in the NoC has its own frequency island. The control volume contains all one hop neighbors of the considered router. It includes L1 and L2 links to count injected flits from cores and cache memory.

wormhole routers, where each packet is split in flits that are the atomic transmission units for the NoC. In order to enhance the readability we define four quantities as follows:

- *power metric and actuator* - We define the router *frequency* as the proxy for power in our model, due to the well known formula that states the linear relation between frequency and power as follow:
$$P_{dyn} = \alpha C V^2 f \qquad (1)$$
where P_{dyn} is the dynamic power of the router, C represent the equivalent capacitance, V the applied voltage and f the frequency. Also, the router frequency is the control input that can be modified at run-time to optimize power. Note that we do not consider static power, since we are focusing on a DFS actuator.
- *performance metric* - Given a router j, we define the per router congestion metric ($C_{i,j,t}$) as the buffer utilization for router i due to flits that want to go to router j at time t. It is worth noticing that we can define the *local congestion* of a router i as $C_{i,i,t}$ that are the flits stored in the buffers of router i. This last definition can be used as a measure for local performance, as reported in [6].
- *control volume (N_i)* - it represents the virtual envelope for a router i that contains all its direct neighbors as reported in Figure 1. The boundary of the control volume is crossed by all the router links connected to the neighbors of router i, as well as the links connecting router i to its CPU and L2 cache.
- *incoming flits $InF_{N_i,t}$* - it is the number of flits that have been inserted in a control volume within a specific time interval.

Starting from the above definitions, one could try to use the local congestion $C_{i,i,t}$ as the performance metric for router i, since a low congestion means low latency due to low conflicts on shared router resources. However, the utilization of the local congestion metric is not enough to model the frequency-congestion relation on a specific router. Consider for example a two routers scenario as depicted in Figure 2. The downstream router (r_d) has an initial state, where frequency is zero and all the input buffers empty, while the upstream router (r_u) has a frequency greater than zero and flits stored at its input ports. As at router r_d the local congestion $C_{r_d,r_d,t}$ is zero, each reasonable optimization policy would not increase the frequency to minimize power consumption. However, r_u has both $C_{r_u,r_u,t}$ and $C_{r_u,r_d,t}$ different from zero, since it has flits that want to go to r_d. In such a case flits can never be received by r_d, resulting in a stall situation.

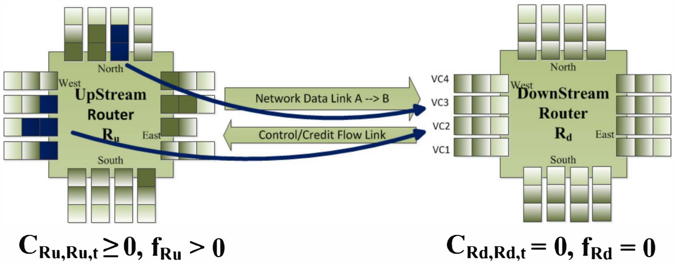


Fig. 2: Local congestion limitations considering up- and downstream router pair, where the upstream one has flits for the downstream one that can be never sent.

In light of this issue, we compute the congestion for a router i counting also the number of flits directed towards that router that are stored in the routers (and CPUs, memory controllers, etc.) one hop before, or, in other terms, the flits contained in the control volume and directed to i. This choice for the definition of congestion also has the advantage of an anticipated response to a sudden increase in the number of flits directed to a specific router to improve the performance of the control schemes. To this extent we define the global congestion for a router i at time t as:

$$C^G_{i,t} = \sum_{j \in N_i} C_{j,i,t} \qquad (2)$$

where $\sum_{j \in N_i} C_{j,i,t}$ represents the sum of the congestions on neighbors of the considered router i due to flits that want to go to router i.

Once a suitable congestion metric has been found, the goal is then to derive a dynamic model of the frequency to congestion relation. A possible way is to start from physical considerations on the operation of a NoC router, and compute a balance of flits in the control volume of a given router. By following such considerations, the following model was obtained:

$$C^G_{i,t} = C^G_{i,t-1} + InF_{N_i,t} - \alpha * f_{t-1} \qquad (3)$$

First of all, the complete model includes saturations not shown here to ease its understanding. Those saturations simply account for the fact that the congestion value can never become negative, nor increase above the sum of all buffer sizes in the control volume.

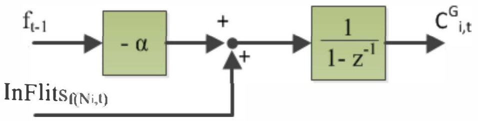


Fig. 3: Dynamic frequency-congestion model. The frequency is a controllable input, while the injected flits is a non controllable input.

The model contains an integrator, which is easily explained considering that if no flits enter the control volume, and the frequency of the router is zero, the congestion remains constant.

Also, physical considerations suggest that flits entering the control volume and directed towards the considered router cause an increase in its congestion, which explains the $InF_{N_i,t}$ term. In the proposed model the injected flits are considered as a disturbance, since they cannot be controlled, while the router frequency is the control input.

The frequency impact on congestion depends on several implementation details, i.e. the internal router structure and the arbiter policies, while it is reasonable to assume that a frequency increase decreases the congestion. Moreover, the impact the frequency has on the congestion is function of the actual traffic pattern. If we define the input or/and output serialization, as a set of flits from a common input port or multiple flits that want to go to the same output port, respectively, the congestion due to serialization increases with higher traffic, where collisions are more frequent. In this perspective, we added an unknown parameter, α in Equation (3), to describe the frequency-congestion model.

This last step needed to complete the model is therefore the identification of the α parameter based on experimental data. For this purpose, it can be noticed the proposed dynamic model belongs to the family of deterministic autoregressive models with exogenous input model (ARX) [15].This allows the use of standard identification techniques to find the α parameter. The resulting ARX model is the one depicted in Figure 3.

B. Simulation framework

The experimental data needed for the identification process was obtained starting from the GEM5 simulator [16], that is an event-driven simulation framework for multi-core architectures enabling an accurate NoC model [17]. The baseline GEM5 has been modified introducing three different main logical blocks.

First, the event management system of the simulator has been extended to support the possibility to change the frequency of NoC routers throughout the simulation, thus mimicking the DFS behavior. In detail, it is possible to specify flexible frequency islands ranging from a per router granularity up to a single global island for the entire multi-core.

Then an implementation within the simulator of a two-way handshake protocol for resynchronization among routers has been introduced. The protocol is similar to the work in [5], and the implementation accounts for the timing overhead of the resynchronizer component.

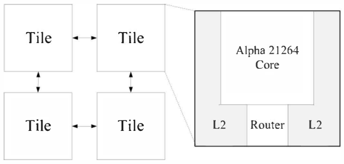


Fig. 4: Tile details for the simulated multi-core.

Last, a flexible logging framework was introduced to collect run-time data. In particular, this allowed to extract the global congestion and InF metrics, as defined above, from arbitrary NoC routers with traffic patterns produced by the simulated cores running applications taken from the MiBench [18] suite.

IV. Results

The accuracy of the proposed model as well as its feasibility and usability issues are discussed in this section. Section IV, describes the experiments that have been performed to collect identification data, while Section IV-B presents the obtained model along with a discussion of its properties.

The data required for identification are obtained from the simulator described in Section III-B. Due to the lack of space, we considered a representative 2D-mesh 16-core tiled architecture only. Each tile is composed of an Alpha 21364 processor running at 2GHz, with private L1 caches and a distributed shared L2 cache composed of 16 banks, one connected to each router as depicted in Figure 4.

A. Simulation setup

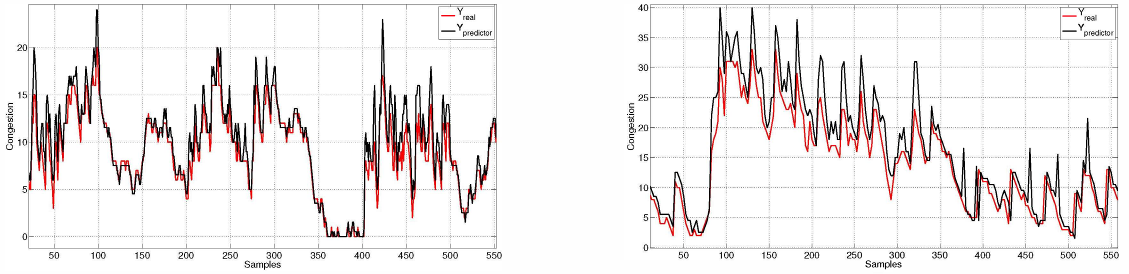
We simulated different scenarios using several applications taken from the MiBench suite.

For each simulation, one of the routers was selected as the target for identification. Its operating frequency was alternated between 2GHz and 100MHz every 500 clock cycles, so as to stress the frequency to congestion dependence, leaving the frequency of all the other routers fixed at the default value of 2GHz. Also, the logging framework was exploited to collect every 5 clock cycles congestion and InF metrics, as described in Section III-A.

The model has been evaluated considering different microarchitectural parameters. In particular data collection and identification has been done considering 2 and 4 virtual channels for each virtual network in the considered multi-core. Moreover, we checked the dependence of the model with respect to the router placement in the NoC. The first half of the experiments was performed by selecting a router at the edge of the NoC, router 0 in Figure 1. This router is connected to two other NoC routers and a memory controller. In the second half of the experiments, router 5 was selected. This is an inner router connected to four other NoC routers and no memory controller.

B. Model identification

The presented model has been completed by identification of the α parameter based on the full dataset obtained by concatenating the traces for all the tested MiBench [18].



(a) Basicmath.

(b) Stringsearch.

Fig. 5: Experimental data (red) and 1 step prediction (black) using the estimated model against different benchmarks for Router 5 considering the 2D-mesh topology presented in Figure 1 and 2 VCs.

TABLE I: Estimated alpha parameter and Final Prediction Error (FPE) for the NoC configuration with two virtual channels.

benchmark	$R_{\bullet}$		R_5	
	FPE	α	FPE	α
basicmath	0.83	0.086	2.63	0.357
stringsearch	17.40	0.308	6.38	0.515
bitcount	19.13	0.309	6.61	0.574
dijkstra	4.68	0.225	3.58	0.173
susan	15.10	0.210	5.91	0.325
qsort	9.02	0.245	8.42	0.343
ALL	4.38	0.213	2.42	0.235

TABLE II: Estimated alpha parameter and FPE for the NoC configuration with four virtual channels.

benchmark	$R_{\bullet}$		R_5	
	FPE	α	FPE	α
basicmath	0.82	0.086	3.04	0.335
stringsearch	17.19	0.297	6.97	0.567
bitcount	15.93	0.269	7.01	0.596
dijkstra	2.20	0.100	2.71	0.160
susan	14.28	0.206	6.13	0.334
qsort	8.38	0.246	7.92	0.334
ALL	2.34	0.101	2.77	0.184

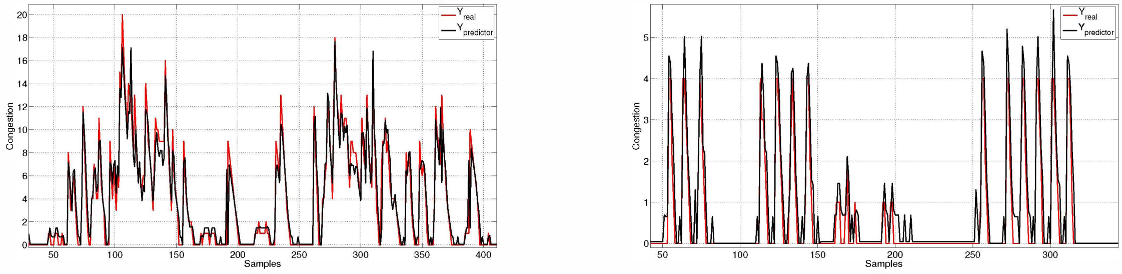
To test the dependence of the α parameter on the application-generated traffic patterns, the identification was repeated using traces from individual datasets. Relevant results are reported in Table I and in Table II for identification with two and four virtual channels respectively. Tables share the same format, the first column reports the benchmark trace used for model identification while the *ALL* row represents the identification result using the concatenation of all the tested MiBench benchmarks. The other rows report selected benchmarks only due to space limitations. Then, the two multi-columns report data for router 0, i.e. R0, and router 5, i.e. R5. In particular each multicolumn reports the Final Prediction Error (FPE) and the α coefficient identified starting from the specific benchmark trace. The Final Prediction Error (FPE) represents a neutral measure for the accuracy of the model to describe data, where lower values mean better model [15].

Starting from data in Table I and Table II three different observations may be discussed. First, the physical placement of the router, i.e. R0 corner-side or R5 central router affects the estimation of the α parameter as reported in Table I and Table II. For example, the first line in Table I reports an estimated α for *basicmath* trace equal to 0.086 on R0 and equal to 0.357 on router 5. This means that a frequency increase has a great impact on an internal router then on a corner-side one. However, the identified α parameter considering all the traces provides similar results for both internal and corner routers. For example, the *ALL* row in Table I reports an α to 0.213 and equal to 0.235 for router zero and router 5 respectively.

Second, the behavior of the single application greatly influences the model identification, since we experienced different FPE values for different applications. For example, considering the same 2 VCs multi-core for router 5, i.e. results in multicolumn *R5* of Table I, the identification using *basichmath* trace has an FPE equal to 2.63, while the FPE for *stringsearch* is 6.38. Moreover, the α coefficients are 0.357 and 0.515, respectively. Thus, the applications seem to influence the frequency-congestion relation even if the coefficient ranges between 0.173 and 0.574 considering R5 in Table I. However, the identified model does not contains saturations yet, that allow for an increased accuracy prediction.

While the variability of the α parameter due to different applications is limited, there is a need for a comprehensive evaluation of the ability of the proposed model to track the experimental data.

In this perspective, we reported another set of experiments, where the identified model augmented with saturations has been used to predict single benchmark traces as reported in Figure 5 for router R5 for the 2 VCs architecture. The model has been identified from all data collected for router R5. Then, we assessed how the identified model can predict the trace of a single application. In particular, we focus on the accuracy, i.e. how well the model approximates data, and tracking property, i.e. the ability of the predictor to follow data with great excursions. We also assess this property considering R0 data traces, while prediction results are reported in Figure 6. Figure 6b allows to assess the tracking property of the proposed model, since trace (red line) presents high variations



(a) Dijkstra.

(b) Bitcount.

Fig. 6: Experimental data (red) and 1 step prediction (black) using the estimated model against different benchmarks for Router 0 considering the 2D-mesh topology presented in Figure 1 and 2 VCs.

and the one-step predictor accurately mimics such behavior with minimum delay. The same behavior is observed in Figure 6 where the estimated predictor can follow a trace that is highly variable and variations are quite fast. On the other side, Figure 5a highlights the accuracy of the predictor consider a trace with a variable behavior. For example, the congestion dynamic is quite fast between samples 400-500, while it has a slower dynamic in the rest of the plot. Despite of the changing behavior of the trace, the predictor can reasonably follow the simulation data.

V. Conclusions

We presented a simple yet accurate dynamic analytical model to bind the frequency of a router to a congestion metric in a NoC-based multi-core developed starting from physical insights. Our proposal focuses on the run-time frequency tracking allowing to mimic real DFS equipped routers. In particular, the proposed analytical model represents a valid solution to design optimal power-performance policies to manage the router frequency at run-time. The possibility to exploit such trade-off is intrinsic in the model, since congestion represents a performance metric [6] while frequency is a knob to manage dynamic power. Experimental results show the accuracy that the predictor extracted from the model has with respect to different data traces, allowing to use such a model as reference to design power-performance optimization policies. Last, the model validation has been done considering different routers in the NoC, different benchmarks, frequency changes over a wide range and different architectural parameters, i.e. variable number of VCs.

References

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

[2] 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.

[3] I. Miro Panades, A. Greiner, and A. Sheibanyrad, "A low cost network-on-chip with guaranteed service well suited to the gals approach," in *Nano-Networks and Workshops, 2006. NanoNet '06. 1st International Conference on*, 2006, pp. 1–5.

[4] I. Miro Panades and A. Greiner, "Bi-synchronous fifo for synchronous circuit communication well suited for network-on-chip in gals architectures," in *Networks-on-Chip, 2007. NOCS 2007. First International Symposium on*, 2007, pp. 83–94.

[5] A. Alhussien, C. Wang, and N. Bagherzadeh, "A scalable delay insensitive asynchronous noc with adaptive routing," in *Telecommunications (ICT),IEEE 17th International Conference on*, 2010, pp. 995–1002.

[6] 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.

[7] 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 13th International Conference on*, 2012, pp. 229–234.

[8] N. Nikitin and J. Cortadella, "A performance analytical model for network-on-chip with constant service time routers," in *Computer-Aided Design - Digest of Technical Papers, 2009. ICCAD 2009. IEEE/ACM International Conference on*, 2009, pp. 571–578.

[9] U. Ogras, P. Bogdan, and R. Marculescu, "An analytical approach for network-on-chip performance analysis," *CAD-ICS, IEEE Transactions on*, vol. 29, no. 12, pp. 2001–2013, 2010.

[10] 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.

[11] V. Soteriou, N. Eisley, H. Wang, B. Li, and L.-S. Peh, "Polaris: A system-level roadmap for on-chip interconnection networks," in *ICCD 2006.*, 2006, pp. 134 –141.

[12] T. Carlson, W. Heirman, and L. Eeckhout, "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation," in *High Performance Computing, Networking, Storage and Analysis (SC), 2011 International Conference for*, 2011, pp. 1–12.

[13] S. Li, J. H. Ahn, R. Strong, J. Brockman, D. Tullsen, and N. Jouppi, "Mcpat: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *Microarchitecture, 42nd Annual IEEE/ACM International Symposium on*, 2009, pp. 469 –480.

[14] D. Zoni, S. Corbetta, and W. Fornaciari, "Hands: Heterogeneous architectures and networks-on-chip design and simulation," in *IEEE ISLPED'12*, aug. 2012.

[15] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Prentice Hall, Jan. 1999.

[16] 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.

[17] 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.

[18] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown, "Mibench: A free, commercially representative embedded benchmark suite," in *Proceedings of the Workload Characterization, 2001. WWC-4. 2001 IEEE International Workshop*. Washington, DC, USA: IEEE Computer Society, 2001, pp. 3–14.