

Improving the Performance of GALS-based NoCs in the Presence of Process Variation

C. Hernández, A. Roca, F. Silla, J. Flich, and J. Duato
{carherlu, anrope2}@gap.upv.es, {fsilla, jflich, jduato}@disca.upv.es
Universidad Politécnica de Valencia

Abstract—Current integration scales allow designing chip multiprocessors (CMP), where cores are interconnected by means of a network-on-chip (NoC). Unfortunately, the small feature size of current integration scales cause some unpredictability in manufactured devices because of process variation. In NoCs, variability may affect links and routers causing that they do not match the parameters established at design time. In this paper we first analyze the way that manufacturing deviations affect the components of a NoC by applying a comprehensive and detailed variability model to 200 instances of an 8x8 mesh NoC synthesized using 45nm technology. A second contribution of this paper is showing that GALS-based NoCs present communication bottlenecks under process variation. To overcome this performance reduction we draft a novel approach, called *performance domains*, intended to reduce the negative impact of variability on application execution time. This mechanism is suitable when several applications are simultaneously running in the CMP chip.

I. INTRODUCTION

Several years ago chip multiprocessors (CMP) have become the common choice to increase computing power while keeping energy consumption inside the power budget. As the number of cores increases, it is not feasible to interconnect them by using a bus or a crossbar due to scalability concerns, and thus a network-on-chip (NoC) is used. For example, a ring is used in the recent Nehalem-EX processor. In the case of the 48-core prototype by Intel mentioned above, the 24 tiles are interconnected by a 6x4 mesh. This is also the case for the Tile-Gx100 chip [28], which includes five parallel 10x10 bidimensional meshes. In general, both academia and industry agree that NoCs are the best option for interconnecting a high number of cores [10], which will be the common case as VLSI technologies continue leveraging larger integration scales.

CMP chips are feasible because of the large integration scales currently used. Unfortunately, the small feature size of current integration scales also cause some unpredictability in manufactured devices because of process variability. The consequence of process variation is that delay characteristics of manufactured devices do not exactly match the parameters established at the design phase. Actually, variability arises as one of the most important challenges to tackle in the development of new on-chip system architectures as technology scales down from 65nm to 16nm [18].

Several are the sources of variability, like variations in L_{eff} (effective transistor channel length) and also variations in the threshold voltage. These variations are basically a consequence of deviations introduced in the photolithographic process and due to random dopant fluctuations, respectively. Additionally, capacitance and resistance variations introduced as a consequence of varying wire dimensions because of defects in the chemical metal planarization process are another source of uncertainty in current and future chips. Recently, a comprehensive and detailed model for characterizing variability in

NoC links has been proposed [6], clearly showing that links are affected by this phenomenon. This model can be extended to NoC routers in order to analyze how process variation simultaneously affects both components of the network. This analysis would be extremely useful for NoC designers. In this paper we take this challenge and thoroughly analyze the behavior of links and routers in the presence of variability, showing that not all of them can operate at the same frequency.

Having a network where not all of its components are able to switch at the same frequency requires that NoC designers take a decision on how to face this challenge. The most basic approach is to lower the operating frequency for the entire NoC to the frequency of the slowest component. However, when process variation causes large differences in frequencies, this solution may lead to a great performance penalty. For example, in [6], it has been shown that differences in link frequency for a given chip may range from 1.3x in a 45nm technology to 1.8x in a 16nm one. Thus, setting NoC frequency to the one of the slowest component may cause considerable drops in aggregated bandwidth.

A more sophisticated approach to face the fact that not all NoC components are able to switch at the same frequency is allowing that each of them works at its maximum achievable frequency. Such a system is known as GALS system (Globally Asynchronous Locally Synchronous) [12]. The concept behind GALS is quite simple, although implementing such a NoC is not that easy because current design tools are not ready yet for the design of GALS circuits. Nevertheless, GALS systems do not avoid the performance reduction introduced as a consequence of different routers and links switching at different frequencies. As we will show in this paper, slower network components tend to become bottlenecks that create congestion, which slows down network traffic, thus reducing network performance.

The conclusion that can be learnt from the two approaches just presented is that technological solutions to face the different frequencies present in a NoC are not enough, by themselves, to mitigate the impact of process variation in NoCs. On the opposite, some kind of architectural solution is required in addition to the underlying technological approaches. Thus, combining both solutions, the technological and the architectural ones, the impact of process variation on network performance may be drastically reduced. We end this paper by drafting such an architectural approach. This approach, called *Performance Domains*, is useful in the case that the chip is simultaneously executing several applications, assigning some of its cores to each of them. Note that this will be the common case as CMPs increase the number of cores. Virtualization techniques (defining parts of the chip as virtual machines) will also deploy such partitioned scenario.

The rest of the paper is organized as follows. Section II presents previous work in this area and assesses how this paper advances the state of the art in the process variation arena. This section also introduces which are the contributions of this work. Section III presents the NoC that will be used as test bench to apply the new variability model for NoCs introduced in Section IV. Later, in Section V, that model is applied to the NoC designed in Section III in order to assess how process variation affects the components of the network. The results from this section are used in Section VI to analyze the performance of NoCs in the presence of variability. Next, in Section VII, we draft a new approach intended to reduce the impact of process variation in application execution time. Finally, in Section VIII, the main conclusions of this work are presented.

II. RELATED WORK AND CONTRIBUTIONS

Many recent works have already partially analyzed variability in NoCs. For example, in [13] the authors identify the main sources of process variation in NoC links and provide an analytical expression of timing variability from the variation of parameters involved in the interconnect delay. However, this study lacks the influence of spatial features of variability in NoC links. Other studies focus on the impact of within-die variation in devices without considering variations in the links [22][20]. However, links in the NoC are also affected by variability as shown in [13] and [6].

The impact of process variation is also analyzed in [14]. However, this study presents two important drawbacks. First, although the consequences of variability in the router architecture are deeply analyzed, this study does not consider delay variations between routers. Additionally, random variability is inaccurately analyzed because it is simplistically modeled as a percentage of the nominal delay, thus not considering other studies that show that random variations strongly depend on the critical path depth and the size of devices [7] [6]. The second drawback of [14] is that it states that variability will not considerably affect NoC links because it is possible to take advantage of the huge slack present in links in order to compensate delay variations. However, as shown in [6], this is only true when links are designed to present minimal delay at the expense of a tremendous power consumption [4].

As can be seen, there is no study that simultaneously analyzes the impact of variability in routers and links. Nevertheless, the point where most studies agree is that process variation causes that routers in a NoC present different delays, despite that they were initially designed to be identical. The same is true for links. In order to deal with this variety of frequencies in the NoC and thus avoid reducing the frequency of the entire NoC to that of the slowest component, the GALS mechanism could be leveraged. GALS was initially proposed to support mesochronous systems where there is not a single clock signal that synchronously clocks all parts of the system, but they are clocked by independent signals having the same frequency but, probably, different phase. Thus, GALS perfectly supports a NoC where its components operate at different frequencies. Additionally, [16] analyzed how to efficiently build voltage and frequency islands from a GALS-based NoC.

Nevertheless, although the GALS technique enables addressing a system having components working at different

Technode (nm)	45
Link length (mm)	0.83
Core area (mm ²)	0.48
Vdd (V)	0.9
M4-M5 Width (nm)	140
M4-M5 Spacing (nm)	140
M4-M5 Thickness (nm)	280
M4-M5 Height (nm)	290
Dielectric	2.5

TABLE I
PHYSICAL AND ELECTRICAL DATA FOR THE 45NM NoC

frequencies, it is not able to minimize the impact of process variation in NoCs. Actually, this claim is one of the three main contributions of this paper. We will show that GALS-based NoCs present communication bottlenecks caused by the slower routers and links. These slower components cause congestion that spans to the rest of the network, thus reducing performance.

In order to carry out this performance analysis, accurate data about the operating frequency of routers and links in the presence of process variation must be gathered. We have collected this frequency data by synthesizing a entire NoC with a 45nm technology and applying an extension of the variability model in [6] to it. This process variation analysis is another of the contributions of this paper. Moreover, it is noteworthy to mention that, up to our knowledge, this is the first time that such a comprehensive and detailed variability model has been simultaneously applied to all components of a NoC, that is, routers and links. Actually, the variability model presented in this paper is more accurate than previously presented ones, as will be shown later.

The last contribution of this paper is the draft of an architectural proposal for reducing the impact of process variation on the average execution time of applications running in the CMP chip. This proposal, called *performance domains*, is based on assigning regions of the CMP chip to applications considering not only the availability of idle cores but also tacking into account process variation data.

III. DESIGNING A 1.33GHZ NETWORK-ON-CHIP

In order to analyze the influence of process variation on NoC performance, we need a network to be used as a test bench so that the variability model presented in Section IV can be applied to it. Therefore, we have synthesized an 8x8 2D-mesh network using 45nm technology. To do so, we have designed a wormhole router from scratch.

The first concern to address when designing such a network is which will be the CMP configuration the network will be embedded into. To answer this question, a 65nm real implementation NoC layout [15] has been considered. According to this implementation, all cores are identical, and their size is 1mm². Additionally, the gap between cores is 0.2mm. Therefore, links connecting NoC switches are 1.2mm long. In this study, we are going to synthesize an 8x8 CMP using 45nm instead of 65nm. Therefore, core size and link length must be appropriately reduced according to the feature size of the 45nm technology. Table I shows the dimensions for cores and links once scaled down to that technology, as well as other physical and electrical parameters of the 45nm technology, as stated in [25]. In the following sections, the router and links designed to be used in such a CMP are described.

A. Router Architecture

In this section we describe the router design used throughout this paper. The router is a pipelined input buffered wormhole router with five stages: input buffer (IB), routing (RT), switch allocator (SW), crossbar (XB), and link traversal (LT). We have designed a simple router with no virtual channels and five input and output ports. Thus, four ports are intended to provide connectivity with the neighboring routers in the 2D mesh and the fifth port connects to the local computing core. Link width is set to 4 bytes. Flit size is also set to 4 bytes. Input buffers can store four flits. A Stop&Go flow control protocol has been deployed in order to control the advance of flits between adjacent routers. Additionally, the routing stage has been implemented to support the XY routing algorithm. Moreover, there is a RT module for each input port. Similarly, one SW module has been implemented for each output port, although all the five SW modules work in a coordinated way, obviously. Finally, each SW module has been designed using a round-robin arbiter according to [21].

The router has been implemented using the 45nm technology open source Nangate [25] with Synopsys DC. We have used M1-M3 metalization layers to perform the Place&Route with Cadence Encounter. Table II summarizes the frequency and area results of the router implementation.

area / Freq	Prelayout	Postlayout
area (μm^2)	17651	19779
freq (GHz)	1.75	1.33

TABLE II
ROUTER AREA AND FREQUENCY

Table III summarizes the delay, area, and number of gates for each of the modules of the router. Note that the area numbers in that table are for a single instance of each module, but some of them are replicated in the designed router. This has to be taken into account if area numbers in Tables II and III are compared. Additionally, the area numbers in Table III have been obtained by independently synthesizing each module to work at its maximum frequency. When, on the opposite, the whole router is synthesized at once, area numbers slightly change. On the other hand, it is noteworthy to mention that, although the number of gates in the critical path in the XB stage is not the highest one, the gates present in it are slower than the gates present in other stages due to the large load the gates in the XB stage support, thus causing that the XB stage becomes the bottleneck in our router.

module	area (μm^2)	critical path (ns)	gates	critical path depth
IB	3113.45	0.55	177	6
RT	124.26	0.32	72	8
SW	337.88	0.52	35	12
XB	1975.6	0.75	519	8

TABLE III
AREA, DELAY, AND NUMBER OF GATES FOR THE ROUTER MODULES

B. Link Design

When designing a link several concerns must be taken under consideration for the sake of efficiency. More specifically, power and area must be optimized for a target link delay. For example, as links are usually long interconnects, they will present a considerable capacitance and resistance. To deal with

them, repeaters are used. Actually, repeater insertion is an efficient method to reduce interconnect delay and signal transition times. In our case, in order to minimize power consumption [4] we have chosen the proper number of minimum sized repeaters that satisfies the delay constraint imposed by the frequency of the router presented in the previous section. As shown before, the post-layout router delay is equal to 0.75ns. Therefore, our link has been designed to present a delay similar to the delay of the router in order to save power. To satisfy those premises we choose a supply voltage equal to 0.9V and a link consisting of 5 repeaters of size 2. Additionally, links are placed in metalization layers M4 and M5, whose physical dimensions are shown in Table I. With this configuration we obtained a nominal link delay of 0.67ns by using SPICE and the PTM model for 45nm [26] for the link simulations.

IV. CHARACTERIZING VARIABILITY IN NOCs

Variability sources can be divided into front-end and back-end ones. The front-end phase of the IC fabrication process is related with the steps involved in the creation of devices whereas the back-end stage comprises steps involved in the wiring definition. Front-end process variation can be further decomposed into systematic and random components. It is commonly accepted that random and systematic variations are uncorrelated [20]. Additionally, spatial correlation depends on some ρ parameter dependent on the exact manufacturing process. The ρ parameter represents the fraction of the chip that is correlated. Values for the ρ parameter used in this study range from 0.5 (half of the chip size) to 1 (the entire chip) according to [20]. On the other hand, back-end variation sources are capacitance and resistance variations due to defects introduced by the chemical metal planarization process in metalizations, for example. The way back-end variation affects a NoC depends on the exact dimensions of the metalizations considered. However, as shown in [6], when NoC links are designed for minimizing power consumption, delay variation due to back-end variation is less than 0.1% because in that scenario link delay is dominated by repeater delay and therefore variations in wire resistance have a negligible contribution to the resulting link delay. For these reasons, we are not going to consider in this study the influence of back-end variability.

A. Modeling Front-end Systematic Variation

The systematic component of variation is strongly related with the photolithographic process. Lens aberrations may lead to an important systematic spatial non-uniformity of L_{eff} over the reticle field. According to [24], $3\sigma L_{eff}$ can be as high as 12% for 45nm processes.

Additionally, in order to model the spatial non-uniformity of L_{eff} as well as its correlation we have used Gaussian Random Fields (GRF). When using GRF with stationary and isotropic fields, the variance (σ_i^2) of the random field $L(x, y)$, representing transistor gate length (L_{eff}) in the (x,y) die position depends only on the euclidean distance between two given locations [20]. Then, the gate length distribution (L) only depends on a correlation function. The correlation model we have used is the spherical model proposed by [20], which is derived from the measurements of [5]. In this model we have used two different values for the ρ parameter: $\rho = 0.5$ and $\rho = 1$.

In order to be able to catch the spatial features of variability, the layout of the NoC implementation presented in the previous section has been considered to compute L_{eff} maps on it. To do so, the chip surface has been discretized using a 1000x1000 square matrix and R[27] has been used to implement the Gaussian Random Fields with the spherical model mentioned before. Note that the matrix size used allows that each router in the NoC is assigned 467 points of that matrix and each link is represented by 127 points.

Once the L_{eff} values are computed for all the chip surface, the delay for every gate in routers, or repeaters in links, is computed according to the explanation in Section IV-C

B. Modeling Front-end Random Variation

The main source of random variation in NoCs is threshold voltage variation due to Gaussian Random Dopant Fluctuations (RDF). According to [8], RDF will increasingly affect deep submicron technologies scaling below 45nm. We set $3\sigma_{V_{th}}$ variations to 40% according to [24] for a 45nm technology. To compute $\sigma_{V_{th}}$ for a given device, according to [1], we can use Equation 1, and then relate the $\sigma_{V_{th0}}$ value of the minimum size device with the $\sigma_{V_{th}}$ of a device of size h , as shown in Equation 2, which clearly shows that $\sigma_{V_{th}}$ can be minimized by increasing the width of devices, represented by h . However, this would increase the area required to implement the circuit as well as the power consumption, which is not a good option.

$$\sigma_{V_{th}} \propto \frac{1}{\sqrt{W_{eff}L_{eff}}} \quad (1) \quad \sigma_{V_{th}} = \frac{\sigma_{V_{th0}}}{\sqrt{h}} \quad (2)$$

C. Introducing Process Variation into Routers and Links

Once we know the behavior of random and systematic variations we have to introduce both variability sources in our target design. To do so, we will separately simulate the presence of variability in links and routers.

In order to introduce process variations in routers they will first be synthesized using Synopsys DC and placed & routed using Cadence Encounter. Then, after post-layout synthesis, we will get an accurate *.sdf* file representing the delay of each cell of the router and a *.def* file representing the location of a given cell. We will use the *.def* file in order to map the L_{eff} values previously computed with the R tool to individual logic gates at each of the routers in the network. Then, we will calculate the delay of each router cell according to the value of L_{eff} assigned to it and will modify the *.sdf*, which will be used by the PrimeTime tool in order to compute the delay of the router. This methodology was already used in [22]. Note that in our study, delays for different routers are individually computed, thus taking into account their position in the die.

In order to compute gate delay we know from [17] that delay of devices can be represented by Equation 3.

$$D \propto \frac{L_{eff}^{1.5} * V_{dd}}{(V_{dd} - V_{th})^\alpha} \quad (3)$$

$$V_{th_{eff}} = V_{th0} - V_{dd} * \exp(-\alpha_{dibl}L_{eff}) \quad (4)$$

For applying that equation, we need to know V_{th} which will simultaneously depend on systematic and random variations. In order to compute the systematic component of V_{th} , we have to catch the dependence of the threshold voltage with gate

length. This dependence is satisfied with the model presented in [3] and shown in Equation 4. In that equation V_{th0} is the threshold voltage for long channel transistors and α_{dibl} is the DIBL coefficient. Then, applying to those formula parameters for the 45nm technology taken from [7] we get the systematic component of V_{th} .

Regarding the random component of V_{th} , it is obtained by using Equations 1 and 2. Note that we know the exact size for each cell in our design and therefore we will compute individualized random variations for each of them. Actually, this is an improvement over previous studies that incorporated random variability over design netlists because they assume that all gates in the design were equally affected by random variability independently on their size.

The last step is adding the effect of random variation and systematic variation and compute the final V_{th} value for a given device. To do so, the resulting value of the random $\sigma_{V_{th}}$ is added to the systematic $V_{th_{eff}}$ and the result is then used in Equation 3 for asserting the final delay value of the device.

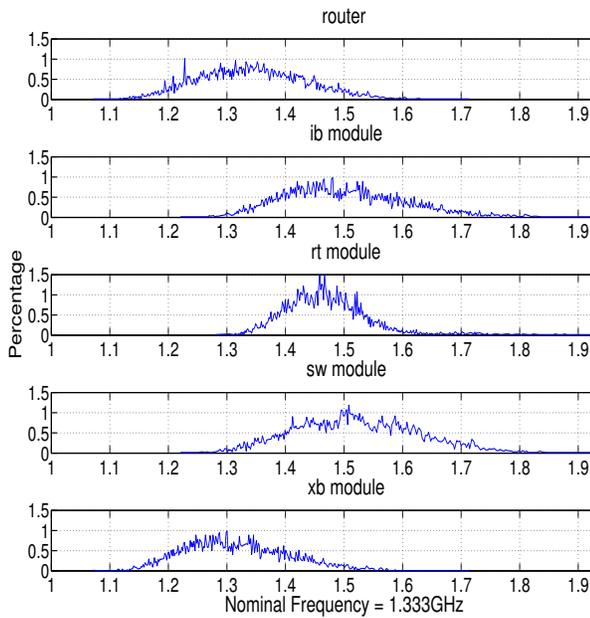
In order to introduce process variations in links, they are simulated with SPICE using the PTM model of 45nm [26] to be able to introduce variations. The methodology to characterize the presence of variations in NoC links is presented in [6]. The location of links is known based on the real NoC implementation presented in [15]. With the location of links and the number of repeaters of the designed link, we know exactly the L_{eff} value of each repeater of each link in the network. As PTM does not directly uses the effective channel length we translate L_{eff} variations into L_{gate} variations. Note that from [23] we know that $L_{eff} = L_{gate} - 2 * L_{int} + X_L$. Where L_{int} and X_L represent the channel length offset due to mask effect. Random variations as in the case of the routers are introduced varying the threshold voltage taking into account the size of repeaters.

V. ANALYZING VARIABILITY IN NOCS

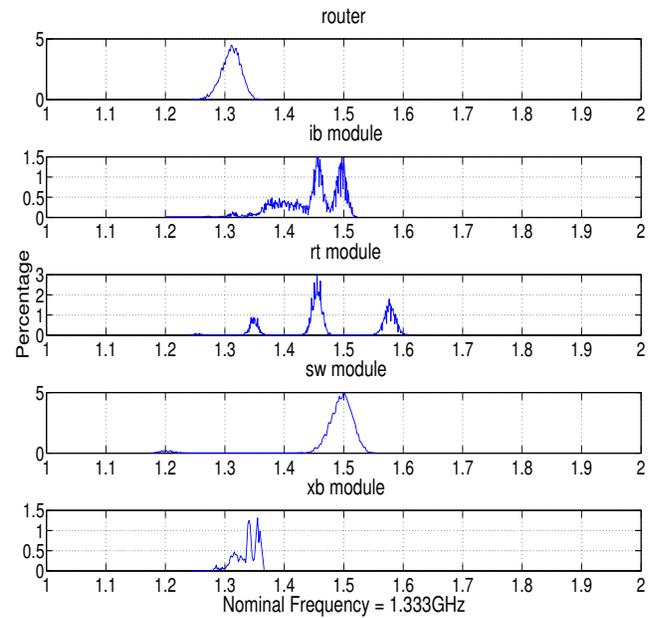
A. Variability in the Router

In order to analyze the effects of variability in the router presented in the previous section, we applied the variability model to 200 instances of our 8x8 mesh NoC synthesized using 45nm technology and study how variability modifies the operating frequency of each of the routers of the network and also each of their modules. From the 200 NoC instance analyzed, 100 of them were produced using a value equal to 1 for the ρ correlation value while the other 100 were produced with $\rho = 0.5$. It is important to remind the reader that the nominal operating frequency without variability is 1.333 GHz. We analyze the influence in the router of random and systematic variations. Figures 1(a) and 1(b) show the probability distribution function (*pdf*) of the operating frequency of the router and each of its stages in two scenarios: when only systematic variation with correlation 1 is considered(1(a)) and when only random variation is taken into account(1(b)). Figures 1(a) and 1(b) show that systematic variation has a larger influence in the operating frequency of the router than random variability.

Additionally, as is shown in figure 1(a) Frequency variations in a router when only systematic variability exists present similar *pdf* for all the modules. This is due to the fact that systematic variability is highly correlated. Thus, as the router



(a) Only systematic L_{eff} variations considered ($\rho = 1$).



(b) Only random V_{th} variations considered.

Fig. 1. Frequency

is small, all the components of the router are affected by variability in a similar way. This means that if the frequency of one of the stages is reduced because of systematic variability then the frequency of the other stages will probably be all reduced. Therefore, it can be seen as a biased variability that causes that the critical path does not change among the router, thus making that the XB stage remains being the bottleneck of almost all instances of the routers.

On the other hand, frequency variation in a router when only random variability is considered does not present similar *pdf* for all the stages. This is due to the nature of random variability which differently affects two adjacent components. Thus, random variability may even be reduced or canceled as the number of gates in a chain of logic increases [6]. Therefore, the critical path of a stage may change depending on the variability map of each router. Thus, the peaks of the *pdf* in figure 1(b) represent a different critical path inside the corresponding stage. This non-biased variability makes that the bottleneck of the router will not always be the XB stage but other stages can constraint the maximum operating frequency of the router. This can be seen in Table IV which shows the correlation between the operating frequency of the stages and the operating frequency of their router. Note that the correlation when only systematic (sys) variability is considered is higher than when only random (rnd) variability is taken into account. Additionally, the correlation of the XB stage is the highest one in all cases, as explained before.

Figure V-A shows the probability distribution function (*pdf*) of the operating frequency of the router and each of its stages when systematic and random sources of variation are simultaneously considered and correlation is $\rho = 1$. Table IV shows that there exist small differences in frequency, between high and low correlation.

Table V shows the main parameters of each configuration described above. It shows the nominal, maximum, mean and minimum frequencies and the frequency variation of each *pdf*.

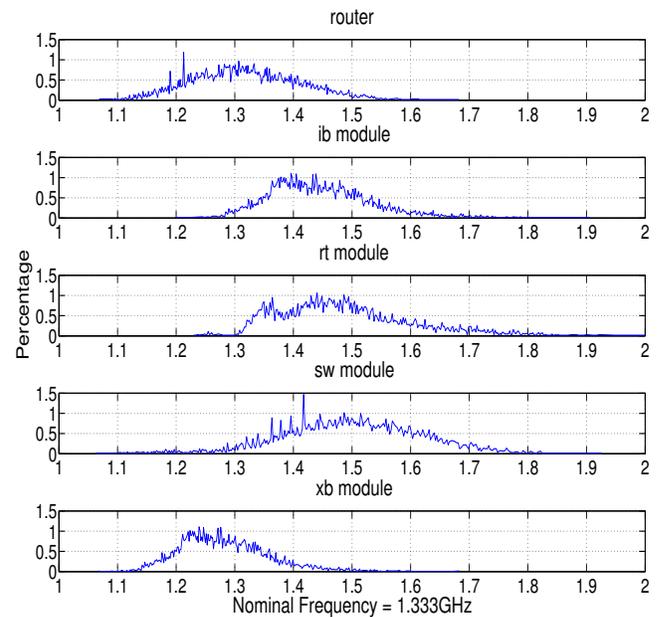


Fig. 2. Frequency variation in router pipeline as a consequence of both systematic and random variations.

Frequency variation is computed as (σ/μ) where σ is the standard deviation and μ is the mean of the *pdf*. Data in Table V confirm that the exact value of the ρ parameter introduces very small differences. Moreover, random variability moves the mean frequency more than the systematic one. As mentioned before, this is due to the fact that random variability makes that the critical path changes from one instance of the router to another more often than systematic variability.

B. Variability in Links

Figures 3, 4, and 5 show link operating frequency variation as a consequence of systematic L_{eff} variation, random V_{th}

stage / variability	sys($\rho = 1$)	sys($\rho = 0.5$)	rnd	sys($\rho = 1$)+rnd	sys($\rho = 0.5$)+rnd
IB	0.9935	0.8941	0.1493	0.9659	0.8318
RT	0.9968	0.7617	0.2867	0.9685	0.7651
SW	0.9945	0.9857	0.0938	0.8896	0.8833
XB	1.0000	0.9997	0.9998	0.9754	0.9876

TABLE IV
CORRELATION BETWEEN STAGE DELAY AND ROUTER DELAY

Parameters	sys($\rho = 1$)	sys($\rho = 0.5$)	rnd	sys($\rho = 1$)+rnd	sys($\rho = 0.5$)+rnd
Nom. Freq.	1.3333	1.3333	1.3333	1.3333	1.3333
Max. Freq.	1.7153	1.6639	1.3661	1.6835	1.6420
Mean Freq.	1.3339	1.3344	1.3101	1.3099	1.3112
Min. Freq.	1.0707	1.0811	1.2422	1.0672	1.0672
σ/μ	0.0672	0.0684	0.0124	0.0681	0.0692

TABLE V
NOMINAL, MAXIMUM, MEAN AND MINIMUM FREQUENCIES AND FREQUENCY VARIATION OF A ROUTER

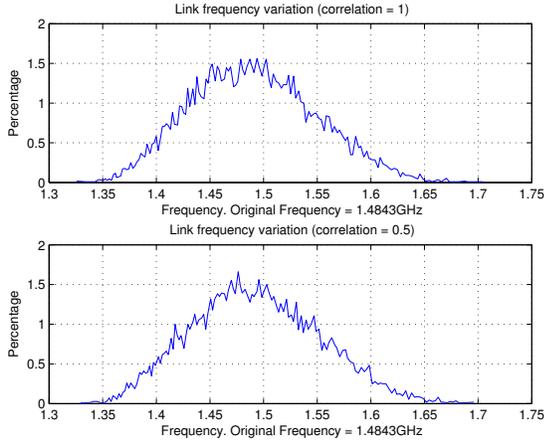


Fig. 3. Link frequency variation as a consequence of systematic L_{eff} variations.

variations, and both random and systematic variations, respectively. Figure 3 shows that when considering only systematic variations, link operating frequency varies between 1.35GHz and 1.7GHz, for both values of ρ , despite that the nominal frequency was 1.48GHz. The exact value of correlation does not introduce significant differences, as shown by the σ parameter ($\sigma = 7.2$ and $\sigma = 7.0$ for $\rho = 1$ and $\rho = 0.5$, respectively). When only random variations are analyzed, variations in wires move practically in the same range than systematic variations. The bottom plot of Figure 4 shows the maximum achievable operating frequency of all wires of each link in the network. However, as all wires of a given link have to work at the same frequency, the slowest wire will cause a considerable operating frequency slowdown. This is shown in the top plot of Figure 4. Where the mean frequency is reduced to 1.34GHz and the frequency variation is reduced to 5.58%. This effect is similar to the behavior analyzed by Bowman et. al in [2]. In this work it is shown that when the number of critical paths increase the mean delay increases and the standard deviation decreases respectively. In the case of links, a higher number of wires per link will cause a higher frequency slowdown but also a reduction in the σ of the link operating frequency. Finally, Figure 5 shows that when random and systematic variations are simultaneously considered the average link operating frequency is reduced as a consequence of random variations. The mean values of link operating frequency are 1.31GHz and 1.32GHz ($\rho = 1$

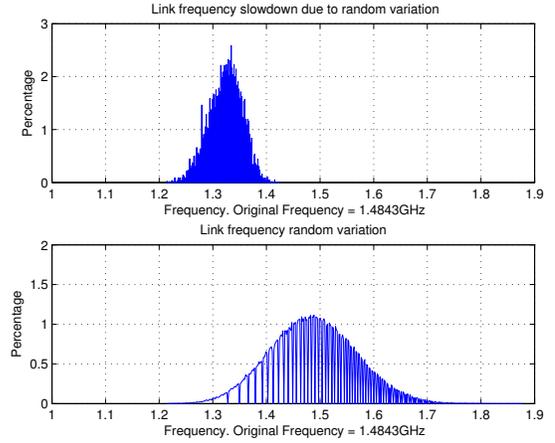


Fig. 4. Link frequency variation as a consequence of random V_{th} variations.

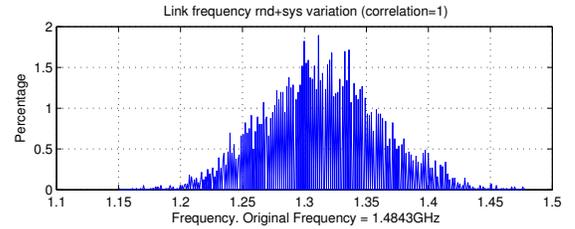


Fig. 5. Link frequency variation for both systematic and random variations. and $\rho = 0.5$). However, frequency variation of links is almost the same than when considering only systematic variations, $\sigma = 7.2$ and $\sigma = 7.5$ for $\rho = 1$ and $\rho = 0.5$, respectively. Note that, due to space limitations, the plot for $\rho = 0.5$ is not shown.

VI. CONSEQUENCES OF VARIABILITY IN GALS-BASED NOCS

In this section we analyze the consequences of variability in GALS-based NoCs. For that purpose, we have simulated the 8x8 CMP test bench network and collected performance metrics, mainly link utilization, message latency and network throughput. The traffic pattern used in intended to simplistically emulate the coherent traffic present in a CMP chip, and is the composition of two different types of traffic. The first one, which accounts for the 60% of the overall network traffic, follows a uniform destination distribution and tries to emulate cache-coherent traffic between cores. The second

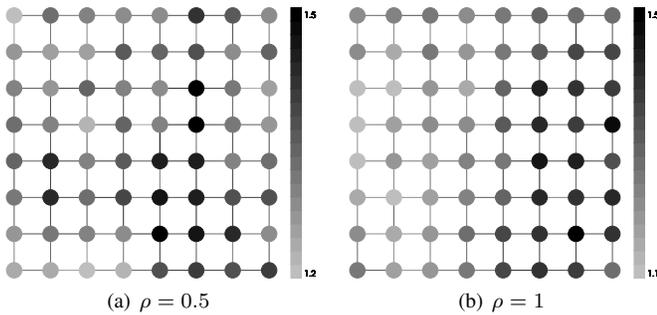


Fig. 6. Operating frequency distribution in a NoC in the presence of PV.

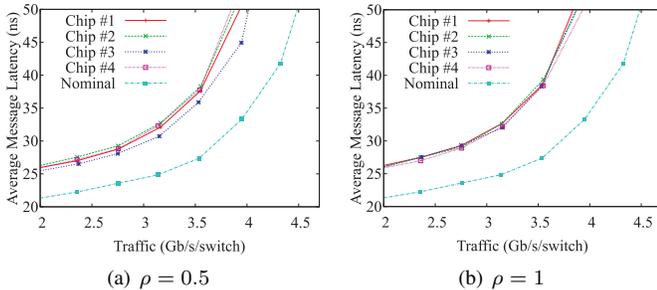


Fig. 7. Network performance in the presence of process variation.

traffic is intended to emulate memory accesses targeted to on-chip memory controllers. We have simulated a CMP chip having four memory controllers located at both sides of the chip: two of them on the right side and the other two on the left side. Additionally, the two memory controllers on the right are directly connected to the 4 upper or 4 lower routers in the right edge of the network. The memory controllers on the left side follow the same interconnection pattern. In this configuration, every core sends 30% of the messages it generates to the memory controller closer to it and the remaining 10% to other memory controllers, with a uniform distribution.

Figure 7 shows the average message latency versus received traffic for several of the chips analyzed. Low and high correlation values are considered ($\rho = 0.5$ and $\rho = 1.0$). As can be seen, in the presence of variability (curves labeled “Chip #n”) the network is able to manage almost 20% less traffic than in the absence of process variation. Moreover, average message latency is increased by 23% even for low traffic loads. On the other hand, it is interesting to notice that the overall performance of the network in the presence of variability is almost independent of the exact characteristics of that variability. This is shown by all the “Chip #n curves” being almost overlapped. Nevertheless, when the correlation of the manufacturing process is lower, we can see more differences in network performance.

One of the reasons for the differences in performance shown in Figure 7 is the lower average network bandwidth caused by link random variation (note that systematic variation does not cause a reduction in the aggregated network bandwidth). However, this reduction in network bandwidth does not completely explain the plots in Figure 7. An important contribution to that performance reduction is shown in Figure 8, that displays utilization for all links in the NoC when it is close to saturation both with and without process variation. As can be seen, process variation causes larger differences in link utilization,

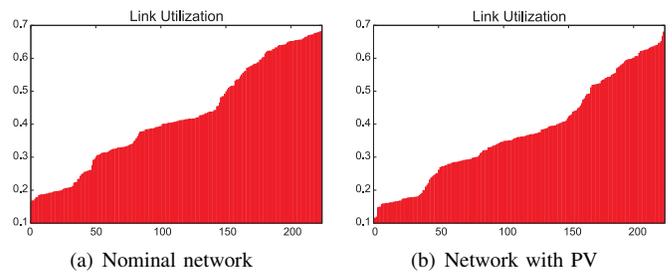


Fig. 8. Link utilization in the presence of process variation.

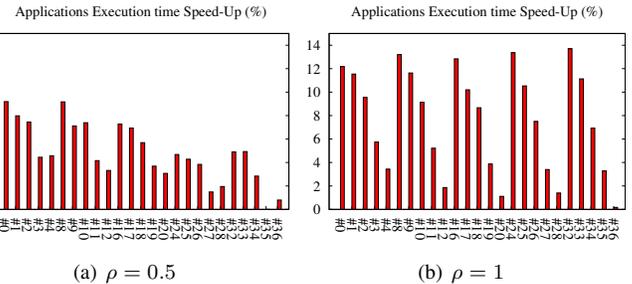


Fig. 9. Speed up when executing freqmine on 9-cores.

as shown in Figure 8(b), where a few links present a much larger utilization than others (on the right end) or a much lower utilization (left end). This uneven distribution of link utilization helps explaining the performance loss in Figure 7.

VII. PERFORMANCE DOMAINS

According to the results in previous sections, the traditional synchronous design technique is not feasible anymore because NoC clock frequency should be lowered to match the frequency of the slowest component in the NoC, noticeably reducing network performance. This fact is widely known by NoC architects. As a solution, GALS systems have been proposed [16]. In these systems, each of the components of the network is clocked at the maximum frequency it allows. However, this technological patch that has several architectural consequences. The immediate one is that slower components quickly become bottlenecks, as shown in the previous section. Therefore, in order to avoid those bottlenecks, we should devise an architectural improvement to current NoCs. This improvement, which we have named *Performance Domains*, is based on two different observations. The first one is that front-end systematic variations cause different delays for different components in the network, but those differences are quite small for adjacent routers and links, due to correlation.

The second observation is that many-core chips are usually devoted to several applications simultaneously, by means of virtualization mechanisms. This virtualization layer assigns cores to applications on demand and usually isolates traffic among the applications so that traffic from one application does not traverse a NoC region assigned to another application. When assigning cores to applications, if no variability data is taken into account, we may assign to an application some cores belonging to a fast area of the chip and some other cores belonging to a slow area. In this case, communication among cores using such an assignment would end up running at the frequency of the slowest router or link because of the bottlenecks mentioned before, causing that the fastest

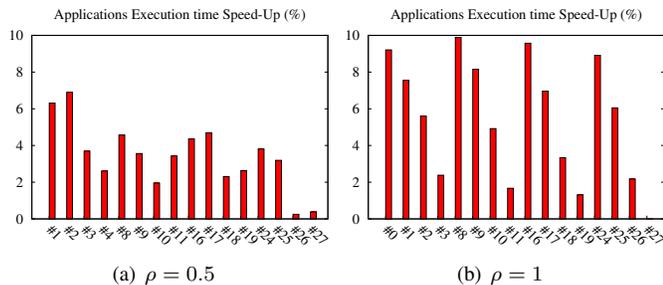


Fig. 10. Speed up when executing swaptions on 16-cores.

resources are underutilized. On the opposite, we propose to assign cores to applications taking into account variability information, thus assigning to a given application cores that are interconnected by links and routers working at similar frequencies. This assignment would not waste faster resources because those resources would not be mixed with slower ones.

As an example of the impact on application time of a variability-aware core assignment, we have simulated the execution of two different applications from the PARSEC benchmark suite on our 8x8 CMP test bench. The first one is the freqmine benchmark running on 9 cores. The second one is the swaptions benchmark running on 16 cores. In the first case we have run the 9-core freqmine program in all the square 9-regions available in the 8x8 CMP shown in Figure 6. In the case for the 16-core swaptions application, we have performed the same test but with square 16-core regions. Obviously, cache coherency between cores is enabled.

Figure 9 shows the speed up for the different executions of freqmine on 9 cores. Labels in the X axis denote the identifier in the 8x8 CMP of the top-left core of the 9-core sub-CMP. As can be seen, depending on the application placement, execution time could be reduced up to 14%. Figure 10 shows the speed up achieved for the swaptions application. In this case, application execution time may be reduced by 10%.

Nevertheless, an efficient variability-aware assignment may be complex to achieve. First, at initialization time, the chip must run a self test to get link and frequency information and use that information in such a way that building a given performance domain later is a straight-forward task that does not consume too many resources. Additionally, when creating a performance domain not only variability data has to be taken into account, but also the performance of the underlying routing algorithm for a given performance domain shape. In this way, performance domains should present a good bisection bandwidth in order to allow for high network throughput. Moreover, as we may end up having non-rectangular domains, the XY routing algorithm is not a valid option anymore. Efficient routing algorithms and routing implementations should be deployed, like the LBDR implementation [19], which additionally allows that routers and links belong to more than a routing region simultaneously. This will be required for accessing the on-chip memory controllers, which will probably be located outside the performance domain.

VIII. CONCLUSIONS

This paper presents a variability model that simultaneously considers process variations in routers and links. We have applied that model to an 8x8 mesh NoC implemented in 45nm

technology, collecting data on how variability affects routers and links and also the individual stages of routers. Additionally, we have analyzed how the diversity of frequencies present in the network due to process variation affects network performance. The conclusion of this analysis is clear. Average application execution time may be improved if the cores that applications use are carefully selected taking into account variability data and not only considering the availability of idle cores. However, additional work has still to be done to make our proposal, called *performance domains* a reality.

ACKNOWLEDGMENT

This work was supported by the Spanish MEC and MICINN, as well as European Commission FEDER funds, under Grants CSD2006-00046 and TIN2009-14475-C04. It was also partly supported by the project *NaNoC* (project label 248972) which is funded by the European Commission within the Research Programme FP7.

REFERENCES

- [1] A. Asenov et al., "Intrinsic threshold voltage fluctuations in decanano MOSFETs due to local oxide thickness variations", IEEE TED 2002
- [2] K. Bowman et al., "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration" in IEEE JSSC 2002.
- [3] Yu Cao et al., "Mapping statistical process variations toward circuit performance variability: an analytical modeling approach", in DAC 2005.
- [4] G. Chen et al., "Low-power repeaters driving RC and RLC interconnects with delay and bandwidth constraints", IEEE Trans. on VLSI, 2006
- [5] P. Friedberg et al., "Modeling Within-Die Spatial Correlation Effects for Process-Design Co-Optimization", in ISQED, 2005.
- [6] C. Hernandez, F. Silla, and J. Duato "A Methodology for the Characterization of Process Variation in NoC Links", in DATE 2010.
- [7] E. Humenay, D. Tarjan and K. Skadron "Impact of process variations on multicore performance symmetry" in DATE '07.
- [8] Kenyon, C. et al. "Managing Process Variation in Intel's 45nm CMOS Technology." Intel Technology Journal, January 2008.
- [9] S. Kottapalli et al., "Nehalem-EX CPU Architecture", Hot chips 21, August 2009.
- [10] R. Kumar et al., "Interconnections in multi-core architectures: Understanding mechanisms, overheads, and scaling", ISCA, June 2005.
- [11] D. Ludovici et al., "Assessing Fat-Tree Topologies for Regular Network-on-Chip Design under Nanoscale Technology Constraints", in DATE'09.
- [12] I. M. Panades et al., "Bi-Synchronous FIFO for Synchronous Circuit Communication Well Suited for Network-on-Chip in GALS Architectures", in NOCS '07.
- [13] M. Mondal et al. "Provisioning On-Chip Networks under Buffered RC Interconnect Delay Variations", ISQED '07.
- [14] C. Nicopoulos et al., "On the Effects of Process Variation in Network-on-Chip Architectures" IEEE TDSC 2008.
- [15] A. Pullini et al., "NoC Design and Implementation in 65nm Technology", NOCS '07.
- [16] U. Ogras et al., "Voltage-Frequency Island Partitioning for GALS-based Networks-on-Chip", Design Automation Conference, 2007
- [17] M. Orshansky et. al "Impact of Spatial Intrachip Gate Length Variability on the Performance of High-Speed Digital Circuits", in TCAD 2002.
- [18] J.D. Owens et al., "Research challenges for on-chip interconnection networks", IEEE Micro, 2007.
- [19] S. Rodrigo, J. Flich, J. Duato and M. Hummel, "Efficient Unicast and Multicast Support for CMPs", in MICRO 2008.
- [20] S. R. Sarangi et. al, "VARIUS: A Model of Parameter Variation and Resulting Timing Errors for Microarchitects", IEEE TSM, 2008.
- [21] E. S. Shin et al. "Round-robin Arbiter Design and Generation," in ISSS, Oct. 2002, pp. 2-4.
- [22] B. Stefano et al. "Process variation tolerant pipeline design through a placement-aware multiple voltage island design style", DATE '08.
- [23] "BSIM4 6.4 MOSFET manual", available online at "www-device.eecs.berkeley.edu/bsim3/BSIM4/BSIM464/BSIM464_Manual1.pdf".
- [24] "International Technology Roadmap for Semiconductors", 2007 Edition, available online at <http://www.itrs.net/Links/2007ITRS/Home2007.htm>
- [25] "The Nangate Open Cell Library", 45nm FreePDK, available online at <https://www.si2.org/openeda.si2.org/projects/nangatelib/>.
- [26] "Predictive Technology Model", available online at <http://www.eas.asu.edu/ptm>.
- [27] "R: A language and environment for statistical computing. R Foundation for Statistical Computing", URL <http://www.R-project.org>.
- [28] "TILE-Gx Processors Family", available at <http://www.tiler.com/>