A new mechanism to deal with process variability in NoC links

Carles Hernández, Federico Silla, Vicente Santonja, and José Duato
Parallel Architecture Group
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022-Valencia, Spain
carherlu@gap.upv.es, {fsilla,visan,jduato}@disca.upv.es

Abstract

Associated with the ever growing integration scale of VLSI technologies is the increase in process variability, which makes silicon devices to become less predictable. In the context of network-on-chip (NoC), this variability affects the maximum frequency that could be sustained by each wire of the link that interconnects two cores in a CMP system.

Reducing the clock frequency so that all wires can properly work is a trivial solution but, as variability increases, this approach causes an unacceptable performance penalty. In this paper, we propose a new technique to deal with the effects of variability on the links of the NoC that interconnects cores in a CMP system. This technique, called Phit Reduction (PR), retrieves most of the bandwidth still available in links containing wires that are not able to operate at the designed operating frequency. More precisely, our mechanism discards these slow wires and uses all the wires that can work at the design frequency. Two implementations are presented: Local Phit Reduction (LPR), oriented to fabrication processes with very high variability, which requires more hardware but provides higher performance; and Global Phit Reduction (GPR), that requires less additional hardware but is not able to extract all the available bandwidth.

The performance evaluation presented in the paper confirms that LPR obtains good results both for low and high variability scenarios. Moreover, in most of our experiments LPR practically achieves the same performance than the ideal network. On the other hand, GPR is appropriate for systems where within-die variations are expected to be low.

1 Introduction

Current and future silicon technologies allow designers to include a high number of cores in the same die, leading to what is known as CMP (Chip MultiProcessor). When the number of cores in the die is high, it is not feasible to interconnect them by using a bus, but a network-on-chip (NoC) is required for the sake of scalability [6].

Unfortunately, current integration scales do not only bring several billion transistors in a single die working at gigahertz frequencies. New challenges have to be faced by designers because of the process variability introduced as a consequence of shrinking transistor size. Several are the causes for variability. On one hand, the field-effect-transistor (FET) speed significantly varies across the die. Additionally, different voltage offsets across the die due to differing average power density and static droops are difficult to estimate at design time and cause variable behaviour. Furthermore, printing transistors on the die is becoming increasingly harder because of their extremely small size. This makes that implemented devices differ from the initially designed ones.

Process variability makes silicon devices to become increasingly less predictable, forcing chip designers to use techniques to avoid losing performance and keeping yield. The first technique used to guarantee the proper operation of the chip against process variability was reducing the clock frequency so that all the parts of the chip can properly work. Unfortunately, this low-cost technique is not useful as variability increases because of the large performance penalty. Therefore, new techniques that adapt to the exact conditions of the manufactured chip are required.

In this paper we focus on overcoming the effects of variability on the links of the NoC that interconnects cores in the chip. The probability of having faulty links might considerably increase in future CMP systems, expected to be implemented with 22nm technology by 2015 [9] due to the great variability caused by the much smaller transistor size and the huge number of links present in the network. More precisely, in this paper we present a new mechanism that adapts link operation to the real conditions of the manufactured chip and therefore it is able to keep links working in the presence of variability. The benefits of such a technique
are twofold. On one hand, such a variability-aware mechanism makes that chip performance does not decrease significantly. On the other hand, yield is maintained. Otherwise, an important fraction of the manufactured chips should be discarded.

The remainder of the paper is organized as follows: in section 2, we introduce the fault model we have considered. Related work is addressed in section 3. Our proposal to deal with the presence of faulty wires in NoC links is described in section 4, while it is evaluated in section 5. Finally, section 6 draws some conclusions.

2 Fault model

As mentioned before, device variability arises as one of the most important challenges to tackle in the development of new on-chip system architectures. Variability is caused by deviations introduced in the fabrication process and by intrinsic effects like atomic-level differences due to small variations in dopant levels. In both cases, process variation affects the delay of a signal propagating between two points in the die. This delay variation, and consequently the maximum operation frequency, can be characterized by a Gaussian probability density function [2].

In this work we assume that, as a result of delay variations, there may be differences in the maximum operating frequency of each wire of a link. This variation becomes quite significant in NoC systems due to the large number of wires that form a link and the increasing number of links in the entire chip because of the increasing number of interconnected cores in the same die. Note that network links tend to use top-most metal layers whose width is several times lambda, and therefore it would seem that links would be the least affected by process variation which causes changes in their physical dimensions. However, as NoC links use to be placed in the top metal layers, the probability of faults may increase because active devices located at the silicon surface have to reach the top metal layers and therefore they use a higher number of vias and metallization layers. On one hand, increasing the number of metallization layers involves increasing variability because the chemical metal planarization process is repeated to build-up all metallization layers [7]. This process is one of the major sources of timing variability as it causes surface imperfections in the interconnect because of dishing and erosion. On the other hand, active devices located at the silicon surface have to reach the top metal layers by using vias. Vias are becoming an increasing concern, as vias and contact resistances become a large portion of overall wiring resistance. Via resistance variation is dominated by the properties of the filling mechanism and the material used [8]. Other important source of variability, in NoC links, is the Leff of the transistors used in link buffers.

When designing the CMP chip, the maximum theoretical operating frequency of the links ($f_{max}$), obtained through the worst-case delay analysis, is represented by a Gaussian probability density function that depends on the parameter variation of the manufacturing process. Therefore, the actual operating frequency targeted at design time ($f_{clk}$) must be lower than the mean of this density function ($f_o$), since not all wires would be able to reach $f_o$ as a consequence of process variations. Actually, $f_{clk}$ must be low enough to guarantee that all the wires in the link are able to switch at that frequency. This behaviour is better shown in Figure 1. The X-axis in Figure 1 represents the operating frequency, while the Y-axis is the probability density function of the maximum operating frequency. It can be seen in the figure that, due to variability, not all wires of the link will be able to switch at $f_o$, but will end showing some differences. The highest probability corresponds to the mean, $f_o$. Additionally, some wires can work at higher frequencies, while others have a maximum operating frequency lower than $f_o$. Therefore, if we set $f_{clk}$ to be equal to $f_o$, half of the wires will not be able to operate properly and, thus, they will be marked as faulty wires. In Figure 1, several Gaussian density functions corresponding to different process variations are shown. The narrowest curve is associated to a fabrication process with low frequency variation, $\sigma$ ($\sigma$ stands for the standard deviation). In order to ensure the absence of faulty wires, the operation frequency, $f_{clk}$, is chosen according to a 6$\sigma$ criterion, i.e. $f_{clk} = f_o - 3\sigma$. However, as process variability is becoming higher, the shape of the probability density function of the maximum operating frequency becomes wider too, that is, as integration scales shrink down transistor size, the curves that describe variability become

![Figure 1. Example of several different Gaussian density functions used for choosing the target operating frequency](image-url)
shorter and wider, leading to increased device variability. Consequently, as variability grows, a lower value of \( f_{\text{clk}} \) must be used in order to keep all the links within its operating range. In these cases, frequency must be noticeably reduced if a 6σ criterion is used, as shown in Figure 1. This frequency reduction keeps yield but considerably reduces performance. In this paper, we propose a new technique to overcome this variability without the need of reducing performance at the same time that yield is maintained.

3 Related work

The simplest way to deal with variability is either discarding the chip because of one or more faulty links, or reducing the clock frequency of the whole chip, thus marketing lower performance and cheaper versions of the chip. A slightly more sophisticated solution is using a fault-tolerant routing algorithm to keep the chip working. However, this approach also reduces chip performance because some links are labelled as faulty just because they have a few slower wires. More sophisticated approaches that keep these links working are required.

Several studies have been made on providing data reliability in NoC links. However, most of them are focused on handling transient errors. These errors are caused mainly by noise, crosstalk effects and electromagnetic interferences (EMI). In order to deal with crosstalk, [10] suggests the use of Crosstalk Avoidance Coding to improve reliability in NoC links. Nevertheless, as transistor size is drastically shrinking, permanent errors are becoming more significant in chip interconnects. For this reason, new techniques able to overcome the presence of permanent errors have to be introduced. In this way, [4] proposes the inclusion of spare wires in each link. Thus, if a link contains some faulty wires, it is possible to choose a complete set of non-faulty wires using a crossbar to perform the transmission. Later in [5], the use of spare wires is combined with other fault tolerant methods such as Hamming codes, time redundancy, and self-timed signaling handshake. That provides the link with the capability of tolerating both permanent and transient errors. On the other hand, [11] corrects timing errors. These errors can be transient or permanent and are mainly caused by interferences, coupling, noise, or variability.

The approaches previously mentioned are able to deal with the presence of both transient and permanent errors. However, their fault tolerance is bounded by design to a limited number of faulty wires per link, although, they do not limit the amount of faulty links. On the contrary, the mechanism proposed in this paper is able to mitigate the effects of variability even if all the wires of the link but one are faulty. This mechanism is focused on solving the problems caused by delay variability in NoC links. Furthermore, it is possible to combine this solution with other mechanisms in order to provide reliability to face transient errors.

4 Dealing with variability

As mentioned in the previous section, our goal is to deal with links where not all wires are able to work at the original frequency. Such a link is shown in Figure 2(a). Figure 2(b) shows a link composed of five wires working at the frequency targeted at design time, \( f_{\text{clk}} \). This is a non-faulty link. Figure 2(b) shows a faulty link. In this link, wires 1 and 4 are not able to switch at the original \( f_{\text{clk}} \) frequency. Wire 1 is slightly slower than the design frequency (90% of \( f_{\text{clk}} \), approximately) while wire 4 switches at less than half of the initial frequency (40% of \( f_{\text{clk}} \)).

The link in Figure 2(b) would usually be labeled as a faulty link at system initialization so that its use is precluded. If this link interconnects two switches of the NoC, a fault-tolerant routing algorithm [1] would be required in order to keep the network working. However, some bandwidth could still be retrieved from this faulty link if it is not discarded. In fact, if the original link shown in Figure 2(a) has an aggregated bandwidth \( B_a = 5 \times \text{bps} \), where \( \times \) is the targeted bandwidth of each wire, the faulty link in Figure 2(b) is still able to deliver approximately \( B_b = 3 \times +0.4 \times +0.9 \times = 4.3 \times \text{bps} \) representing 85% of the initial bandwidth. In real NoC links composed of 128 or 256 wires, for example, having a few slower wires would mean that almost 100% of the bandwidth is still available. Thus, discarding the entire link means wasting bandwidth.

There are multiple possibilities to retrieve bandwidth from faulty links, although the initial solution is basically reducing the frequency of the link to operate at the frequency of the slowest wire [12], as mentioned in the previous section. This option allows the link to be operational at the expense of a reduction in performance. In fact, as the frequency of the whole link is reduced to match that of the slowest wire, the reduction in performance could be noticeable. In the case of the 5-wire link shown in Figure 2(c), the available bandwidth retrieved with this technique would be \( B_c = 0.33(5 \times) = 1.65 \times \text{bps} \). Note that this important reduction in bandwidth would also be present in real and wider links, because the whole link reduces its frequency independently of the number of wires.

In this paper we propose a better technique to retrieve the bandwidth still available in faulty links. This technique would

\[1\quad 4.3 \times \text{bps} \text{ is computed as follows: 3 non-faulty wires provide } 3 \times \text{bps. Additionally, one of the faulty wires provides } 0.9 \times \text{bps and the other faulty wire contributes with } 0.4 \times \text{bps. Therefore, the aggregated bandwidth is } 4.3 \times \text{bps.}\]

\[2\quad \text{Actually, all the wires in the link, shown in Figure 2(c) should work at the slowest frequency, that is, at } 0.4 \times, \text{ and therefore, the aggregated available bandwidth would be } 0.4 (5 \times) = 2.0 \times \text{bps. However, in order to switch wires synchronously with the clock, the operational frequency should be reduced to one third}.\]
is based on discarding the wires that are not able to operate at the original frequency. This idea is shown in Figure 2(d), where wires 1 and 4 are not used. In this case, link bandwidth is reduced to $B_d = 3 \times \text{bps}$, that corresponds to 60% of the original bandwidth. When real links with 128 or 256 wires are considered, between 90% and 95% of the initial bandwidth is still available. Obviously, in order to transmit information properly, flits must be suitably sliced and their bits sent across the non-faulty wires. Additional hardware is required at the transmitter to slice flits according to the available wires and to allow the transmission of bits belonging to two consecutive flits during the same clock cycle, if necessary. Also, additional hardware is required at the receiver to retrieve the original flits. This technique is called Phit Reduction. We will refer to it as PR.

In order to implement this mechanism, two possibilities arise. The first one, hereafter named Local Phit Reduction (LPR), is oriented to fabrication processes with very high variability, that is, where the differences in the number of faulty wires for different links are high. This approach requires the inclusion of the transmitter and the receiver hardware modules in every router port, as shown in Figure 3. This figure shows a typical wormhole router architecture and the location of the hardware modules needed to provide variability tolerance, labeled as HW. Notice that using the LPR approach, all hardware modules must be able to slice flits and to transmit them across the non-faulty wires, and later reconstruct them. Thus, this mechanism is costly in hardware but provides high performance despite of the presence of faulty wires, because transmission across each link is performed using the maximum available bandwidth.

The other way of implementing this technique is named Global Phit Reduction (GPR). When process variability uniformly affects the whole chip, so that differences in the expected number of faulty wires across the links are low, the inclusion of a hardware module to slice and rebuild flits at every port of the network wastes silicon area. In this case, reducing the phit size for the whole network is a better solution. In order to perform this reduction, it is necessary to identify the link with more faulty wires. To do so, an initialization algorithm is run across the network to find such a link. Once this link is found, it will bound the phit size for the whole network to the number of its non-faulty wires. This is accomplished by configuring all the network interfaces so that they adjust transmission to that phit size. Consequently, the number of hardware modules needed is considerably reduced as they are only required at the elements able of injecting and extracting traffic to and from the network. Figure 4 shows an example of a 4x4 mesh network with 3 faulty links. In this case, the initialization mechanism would fix the phit size to 120. Thereafter, all transmissions are performed in a 120-bit phit basis. Only
network interfaces sending packets have to slice flits, and only end receivers have to rebuild flits, considerably saving silicon resources with respect to the LPR approach because the hardware modules located at the external links are quite simple. These modules just have to allow phit bits to use non-faulty wires. However, the use of non-faulty links is not completely optimized and some of the available bandwidth is wasted.

5 Performance Evaluation

In this section we evaluate the proposed mechanism and compare it with the technique that reduces clock frequency. We perform this evaluation in two steps. First, we evaluate the performance of both mechanisms in the context of transmitting information across a single link. Second, we consider the impact that such mechanisms have on the performance of the entire network.

5.1 Link performance

In order to know the link performance of the PR mechanism, several experiments considering different link characteristics have been carried out. Experiments assume that the number of faulty wires are 2%, 5%, 10%, 20%, and 50% of the total wires. Considered link widths are 32, 64, 128, and 256 wires, although only results for a 128-bit wide link are presented because of space limitations. Results for the other link widths are similar.

Throughput achieved by the PR mechanism is shown in Figure 5. This figure shows the throughput achieved by a single link for several injection rates. Achieved throughput is compared against an ideal link with no faulty wires. As shown in the figure, the PR approach is able to reach high throughput when the number of faulty wires is low. More concretely, in the case of having 2% faulty wires, the maximum throughput is 98% of the original one. As can be seen in the figure, as the number of faulty wires increases, the throughput of the link decreases, as expected. When 50% of the wires in the link are faulty, throughput decreases down to 50% of the ideal one.

Figure 6 compares the maximum throughput achieved by the PR mechanism with the frequency reduction technique, hereafter referred to as FR, when injection rate is equal to 1 flit per cycle. As in the case for Figure 5, shown results are related to the throughput of a single link. The PR mechanism achieves the best results, especially when variability is high. On the other hand, the FR approach obtains worse results because in order to ensure the absence of faulty wires, the frequency reduction required is high.

The results presented in this section correspond to the performance of a single link. In this case, there are not differences between the two mechanisms proposed for phit reduction, LPR and GPR.

5.2 Network performance

In this section, we evaluate the impact of our proposal on network performance. Instead of analytic modeling, simulation was used to evaluate the proposals. Our simulator models the network at flit level. The evaluation methodology used is based on the one proposed in [3]. The most important metrics are latency and throughput. Latency is measured in clock cycles. Throughput is the flit reception
rate, measured in flits per cycle per switch.

5.2.1 Simulation environment

The network is composed of a set of switches. The network topology chosen is a 2D mesh as it seems to be the most suitable topology for on-chip networks. Network size is 64 switches (an 8x8 2D mesh) and the width of all links is 128 bits. Switch architecture was shown in the Figure 3. All switches are identical and have 5 ports (4 ports for connecting to other switches and 1 port for connecting to the local processor). Input ports can store 3 flits, while output buffer size is 2 flits. We have modelled a wormhole router where routing and output link selection is performed in one cycle. We also assumed that one cycle is required to transmit one flit across the crossbar and another cycle is required by signals to propagate along the link. Additionally, the Phit Reduction hardware modules require one cycle in both transmission and reception stages.

We have run simulations with synthetic traffic. For each simulation we considered that average message generation rate is constant and the same for all nodes. Elapsed time between two consecutive messages is generated using a uniform distribution. Once the network has reached a steady state, the flit generation rate is equal to the flit reception rate (traffic). We have evaluated the full range of traffic, from low load to saturation. With respect to message destination, we have considered that it is randomly chosen among all the nodes in the network. Message length is constant and equal to 5 flits.

5.2.2 Results

In this section, we present the impact of our PR variability-aware mechanism at network level by showing the performance achieved by the network. Next figures show the performance comparison when using our proposal, both in LPR and GPR modes, and when the operation frequency is reduced. The frequency reduction has been considered both globally to the entire chip (GFR) and locally to the faulty links (LFR). Additionally, an ideal network with no faulty links has been included in the comparisons in order to show the performance loss of the different mechanisms. This ideal network has been labeled as ideal in the figures.

Variability has been modeled as explained in Section 2. We have considered that frequency may suffer two different variations: either 10% or 25%. Table 1 shows the wire distribution for a faulty link. As can be seen in that table, for a variability of 10%, a 128-bit wide link has 79 wires able to switch at the design frequency (f_{clk}), 49 wires switching at half the original frequency, and zero wires switching at one third of that frequency. In the case for the 25% variability, data presented in the table shows a more degraded link. Furthermore, we have considered that not all links in the network are faulty. We have conducted several experiments by setting the amount of faulty links to 5%, 10% and 20%. This allow us to characterize the different mechanisms in different scenarios and to analyze how variability affects the chip links from a local or global perspective.

<table>
<thead>
<tr>
<th>VAR(%)</th>
<th>f_{clk}</th>
<th>f_{clkh}</th>
<th>f_{clkt}</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>79</td>
<td>49</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>70</td>
<td>56</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1. Faulty wire distribution for the variability considered

Figure 7 shows the network performance when faulty links represent the 10% of the total links in the network, the routing algorithm is deterministic and two virtual channels are available for every link. The goal of this picture is to show how an increasing variability affects the network performance. As shown in the figure, all mechanisms are closer to the ideal performance when variability is low (Figure 7a), especially the local mechanisms. As can be seen, both global mechanisms waste too much bandwidth. In the case of reducing clock frequency for the entire network, links operate at 50% of the frequency used in the ideal case. When GPR is used, the worst link in the network has 79 non-faulty wires, and thus that is the phit used for all the transmissions. This phit size represents 61.7% of the original link bandwidth. This is the reason why GPR performs better than GFR. The minimum performance degradation is achieved with LPR as it retrieves more bandwidth from the faulty links. In the case of low variability, the LFR mechanism achieves almost the same performance as LPR. When variability is set to 25%, differences in performance are larger. In this case, clock frequency is reduced to one third of the original one, causing a great performance degradation.
penalty. In this case, the aggregated bandwidth in the network is reduced to just 33% of the initial one. In the case of reducing phit size globally, the narrowest link in the network fixes the phit size to 70 bits, representing a 54.7% of the original bandwidth. As can be seen, as variability increases, differences between GPR and GFR become larger. In the case of the local mechanisms, discarding the faulty wires in a per-link basis provides much better performance that individually reducing the link frequency. The reason is that PR mechanisms retrieve more bandwidth from faulty links. Moreover, the LPR mechanism performs almost similar to the ideal case. Message latency is slightly higher for low and medium traffic load. One of the reasons for the higher latency is that we have accounted for an extra cycle delay due to the transmitter and receiver modules. Additionally, some links in the network (the faulty ones) have less bandwidth. Differences between LPR and the ideal one arise for high traffic loads, where LPR shows higher latency and saturates slightly before.

Figure 8 shows how network performance varies when different number of virtual channels are used. Faulty links account for 5% of the overall links in the network. As can be seen in the figure, when the number of virtual channels increases, our proposals, LPR and GPR, improve their performance with respect to the mechanisms that reduce frequency. This happens due to the well known fact that when virtual channels are used, link utilization increases. Therefore, as our approach provides more bandwidth at link level, as discussed in the previous section, the overall network performance of these approaches also increase. Table 2 summarizes this information. This table presents, for each of the mechanisms evaluated, the network performance improvement when virtual channels are increased from one to two, and from one to three virtual channels. As can be seen, performance improvement in the ideal case is 2.1 and 2.8 when using 2 and 3 virtual channels, respectively. In the case of the frequency reduction mechanism, this improvement is never higher than 1.8. As shown in the table, these mechanisms do not scale network performance when increasing the number of virtual channels. This is not the case for the phit reduction mechanism. As shown, they scale much better. In the case of LPR, performance improvement is similar to the ideal case.

This trend can be clearly noticed in Figure 8(f), where the global mechanism GPR achieves a slightly higher performance than LFR, despite that just a 5% of the links are faulty. When the number of faulty links in the network increase, these differences in performance are more noticeable, as shown in Figure 11(b). Moreover, the performance achieved by LPR is always much better than the rest of the mechanisms. In fact, it achieves almost the performance of the ideal network.

Table 2. Performance improvement when using virtual channels

<table>
<thead>
<tr>
<th></th>
<th>Ideal</th>
<th>GFR</th>
<th>LFR</th>
<th>GPR</th>
<th>LPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1→2</td>
<td>1</td>
<td>2.1</td>
<td>1</td>
<td>1.7</td>
<td>1.8</td>
</tr>
<tr>
<td>1→3</td>
<td>1</td>
<td>2.8</td>
<td>1.6</td>
<td>1.7</td>
<td>1.7</td>
</tr>
<tr>
<td>var 10%</td>
<td>2.0</td>
<td>2.1</td>
<td>1.9</td>
<td>2.3</td>
<td>2.1</td>
</tr>
<tr>
<td>var 25%</td>
<td>2.0</td>
<td>2.1</td>
<td>2.1</td>
<td>2.7</td>
<td>2.1</td>
</tr>
</tbody>
</table>

Figure 7. Network performance comparison in presence of variability. Faulty links are 10% of the links in the network. Deterministic routing with 2 virtual channels is used.
Figure 8. Network performance comparison in presence of variability. Faulty links are 5% of the links in the network. Deterministic routing is used.
As the number of faulty links grows, the performance of the global mechanisms approaches the performance of the local ones. This is shown in Figure 9. In this figure, the process variability considered is 10% and two virtual channels are used. It can be clearly seen that the performance achieved by the global mechanisms is independent of the number of faulty links in the network. It only depends on the severity of the process variability. Additionally, as happened when the number of virtual channels increase, the relative performance of GPR increases with respect to LFR. Actually, what really happens is that local mechanisms reduce their performance as the number of faulty links increase, getting closer to the global mechanisms.

In the network scenario that we have considered, several links are faulty as a consequence of process variability. When local variability-aware mechanisms are used, like LFR or LPR, those links are kept working but become a bottleneck because they provide less bandwidth than non-faulty links. A possible solution to mitigate the performance degradation due to the presence of these lower-bandwidth links is using adaptive routing. We have run several experiments with three virtual channels in order to avoid deadlock and increase link utilization. Figure 10 shows the throughput achieved with a 5% of faulty links for both 10% and 25% of \( \sigma / f_0 \) frequency variability. With adaptive routing LPR achieves practically the same performance than the ideal network. On the other hand, LFR performance suffers a considerable degradation when variability is 25% as shown in 10(b). Figure 11 shows the adaptive routing performance when the faulty links are 20%. Obviously, global solutions as GPR and GFR remain the same. However, LFR and LPR increase their performance degradation.

6 Conclusions

Device variability has become an important concern as technologies move towards nanometer designs. In the context of network-on-chip (NoC), this variability affects the maximum frequency that could be sustained by each wire of the link that interconnects two cores in a CMP system. Designing for worst-case in NoC links combined with the frequency reduction assures the absence of faulty wires. But, as variability increases, this approach results in an unacceptable performance penalty.

In this paper, we have proposed a new technique to deal with the effects of variability on the links of the NoC that interconnects cores in a CMP system. This technique, called Phit Reduction, retrieves most of the bandwidth still available in links which contain wires that are not able to operate at the designed operating frequency. More precisely, our mechanism discards these slow wires and uses all the wires that are not able to work at the design frequency. Two implementations have been analyzed: Local Phit Reduction (LPR), oriented to fabrication processes with very high variability, which requires more hardware but provides high performance; and Global Phit Reduction (GPR), that requires less additional hardware but is not able to extract all the available bandwidth.

The performance evaluation presented in the paper confirms that LPR obtains good results both for low and high variability scenarios. Moreover, in most of the experiments that we have performed LPR almost achieves the same performance than the ideal network. On the other hand, GPR is appropriate for systems where process variability uniformly affects the whole chip. Phit reduction techniques have been compared with frequency reduction techniques, both globally applied to all the links of the NoC (GFR) or locally applied to the faulty links (LFR). In both cases, the phit reduction approach outperforms frequency reduction.

7 Future Work

This paper proposes and evaluates a powerful novel technique to deal with the presence of faulty wires in NoC links.
Figure 10. Adaptive routing with 5% of faulty links

Figure 11. Adaptive routing with 20% of faulty links

As for future work, we will extend our variability model in order to catch spatial correlations of variability, therefore increasing the accuracy of our evaluation. In this sense, we will additionally model permanent faults, attending to current and future fabrication issues. Moreover, in order to better known the benefits of our proposal, we will measure its impact on the execution time of actual applications. On the other hand, we will analyze the hardware modules already built to get a power and area efficient implementation of our approach.

Acknowledgments: This work has been jointly supported by the Spanish MEC and European Comission FEDER funds under grants “Consolider Ingenio-2010 CSD2006-00046” and “TIN2006-15516-C04-0X”; by the European Commission in the context of the SCALA integrated project #27648 (FP6).

References