Self-Calibrating Networks-On-Chip

Frédéric Worm, Patrick Thiran, Giovanni de Micheli and Paolo Ienne
Ecole Polytechnique Fédérale de Lausanne (EPFL)
School of Computer and Communication Sciences
CH-1015 Lausanne, Switzerland
{Frédéric.Worm,Patrick.Thiran,Giovanni.DeMicheli,Paolo.Ienne}@epfl.ch

***Abstract*— Networks-on-chip provide an elegant framework to efficiently reuse predesigned cores. However, reuse of cores is jeopardized by new *deep sub-micron* noise effects that challenge reliability of CMOS technology. Moreover, noise margins are further reduced as supply voltage scale down. We advocate that self-calibrating techniques will be needed to maintain acceptable design trade-off between energy, performance, and reliability. As a result, self-calibrating techniques have to be integrated within networks-on-chip. This paper presents a self-calibrating link and discusses qualitatively the problem of controlling adaptively its voltage and frequency.**

I. Introduction

Multibillion-transistor systems-on-chip require the integration of heterogeneous predesigned cores on a single die. This is especially challenging for products with short time-to-market. In order to meet tight time constraints, designers need (a) to reuse predesigned cores, and (b) to be able to interconnect them suitably. The *networks-on-chip* design paradigm addresses these two points by extending reuse to the interconnect itself. Indeed, it provides a customizable network layer that glues all heterogeneous cores. However, reuse of predesigned cores is, paradoxically, complicated by the advance of CMOS technologies. That is, as technologies scale further down, we observe an increasing large spread in electrical parameters. Because traditional CMOS design relies on worst-case assumptions, many circuits will not exploit the full capabilities of new technologies. This phenomenon is illustrated in Fig. 1.

Self-calibrating designs have been recently introduced [9] as an alternative to worst-case characterization of silicon. Instead of relying on over-conservative worst-case assumptions, self-calibrating circuits adjust their operating parameters to in-situ actual conditions. They rely on two key hypothesis, namely (1) the possibility to detect that the circuit is not operating correctly, and (2) improve the circuit reliability at some cost—e.g., energy or latency. Although posing

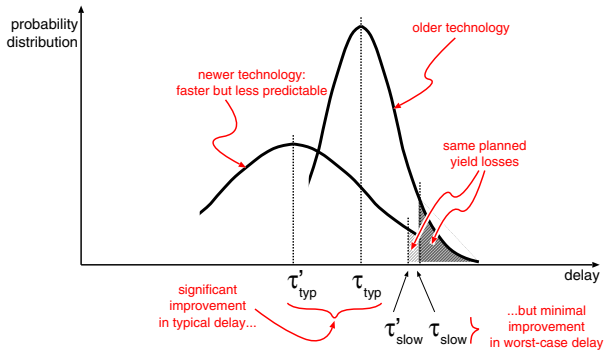

Fig. 1. Delay distribution of a new and an old CMOS technology (the spread of distributions has been exaggerated for the purpose of illustration). Even though typical delay of the newer technology is improved with respect to the older one, we observe barely no improvement on worst-case delay due to the much larger spread of the newer technology.

complex problems, self-calibration has been successfully applied to a processor [1]. In addition, the design of a self-calibrating on-chip link has been thoroughly studied [10]. In particular, the power overhead incurred has been estimated.

We believe that networks-on-chip, as a future communication architecture, cannot ignore self-calibration techniques. In this paper, we discuss the challenges posed by a self-calibrating link with a particular emphasis on the problem of controlling the link voltage and frequency.

A. Outline

Sec. I-B briefly discusses a few concepts related to networks-on-chip. Then, Sec. II expresses self-calibration of a link as a constrained minimization problem. Moreover, we show that the control of voltage and frequency can be decoupled into, respectively, a self-calibrating control problem and a well studied scheduling problem. In Sec. III, we model errors occurring due to over-aggressive operation of the link and explain the requirements of the link encoding scheme. Sec. IV discusses qualitatively the self-calibrating control problem introduced in Sec. II and emphasizes its large potentials. Finally, Sec. V concludes the paper.

B. Related Work

Networks-on-chip borrow concepts from computer networks and apply them to on-chip micronetworks. The communication architecture consists of several layers. Each layer provides a set of functionalities to upper layers. However, layering also entails both a communication and processing overhead, because of the additional information added by each layer. Nonetheless, the whole communication stack can be tailored to application-specific needs by implementing only the required functionalities. The top layer is usually called application layer and, according to the networks-on-chip terminology, consists of *processing elements*. The functional guarantees offered to the application layer are called *quality of service*. For example, a processing element can be guaranteed a certain throughput, maximum end-to-end delay, or a reliability metric (e.g., residual word error rate).

The functionalities offered by each layer and, ultimately, the quality of service guaranteed to processing elements do not impose directly an implementation. That is, the designer can choose any implementation, as long as functionalities of each layer and quality of service are guaranteed. However, different applications may have different requirements. Even more, requirements of one application may vary over time. As a result, instead of designing a communication architecture offering a quality of service meeting the more stringent application requirements, it is desirable to match performance to application needs. This is what *dynamic voltage scaling* techniques enable. They have been first applied to processors [4] [2], studied for chip-to-chip communication links [5], and offer promising results,

0-7803-8834-8/05/$20.00 ©2005 IEEE.

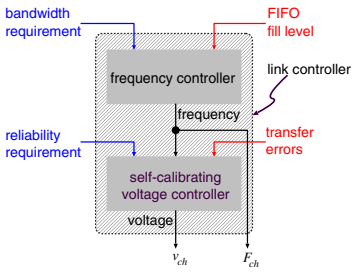

Fig. 2. Selection of the frequency (F_{ch}) and voltage (v_{ch}) required to operate a data link. Based on link information, the top layer determines the frequency required to meet application requirements. This frequency is then input to a self-calibrating voltage controller that determines, for the required frequency, which voltage satisfies the reliability constraint.

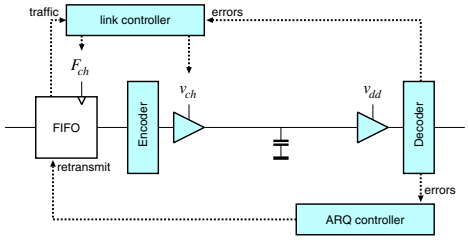

Fig. 3. The link parameter controller determines the voltage v_{ch} and the frequency F_{ch}, based on transfer errors and workload requirements. If the parameters are set too aggressively, or if an error corrupts the data, the decoder raises an error flag causing the ARQ controller to schedule a retransmission.

especially for multimedia workloads that exhibit strong variations in bandwidth requirements.

Dynamic voltage scaling techniques avoid conservative assumptions about application requirements. In that sense, they are complementary to self-calibrating techniques that avoid worst-case assumptions as to silicon capabilities. Applied to communication links, dynamic voltage scaling techniques enable to tolerate uncertainty about bandwidth requirements (i.e., the frequency at which the link is operated), whereas self-calibrating techniques enable to tolerate uncertainty about silicon capabilities to operate at the required frequency (i.e., the voltage required to operate at this frequency). We illustrate in Fig. 2 how dynamic voltage and self-calibrating techniques can be combined in a layered architecture. In what follows, we discuss such self-calibrating links in more depth. In particular, we state an assumption that decouples the control of frequency and voltage, as suggested in Fig. 2.

II. Control of Self-Calibrating Links

We consider a communication link where voltage and frequency are set adaptively by a link controller, as drawn in Fig. 3. On the one hand, the encoding scheme has to detect errors resulting of over-aggressive link operation, or caused by additive noise. These requirements are challenging, since the link controller may cause bit error rates as large as 1, which renders the problem quite unique. On the other hand, the *Automatic Repeat reQuest* (ARQ) and link controllers ensure that the system behaves correctly. The former guarantees an in-order data delivery, while the latter sets link voltage and frequency according to bandwidth and reliability requirements.

The link control problem can be formulated as a constrained minimization. Indeed, we would like to minimize the average energy per information bit under two "quality of service" constraints related to performance and reliability. Regarding performance, we consider the guarantee of the average transfer latency, which is equivalent to guaranteeing a certain throughput. Before formulating the problem, we need preliminary definitions. We call a *word* a group of N bits sent over the link. We denote by E_{w} the energy needed to send a word and by Ξ_{w} the number of cycles needed to send a word through the link. E_{w} can be modelled accurately [6]. Besides other components, it is proportional to v_{ch}^2 and to the word size N. A word may be retransmitted due to transfer errors. We call n the random variable that counts the number of failed transfer attempts. Lastly, we denote by E_{b} the energy needed to transmit an information bit and write

$$E_{\text{b}} = \frac{E_{\text{w}} + E_{\text{rtx}}}{K}, \tag{1}$$

where E_{rtx} is a random variable that describes the energy consumed by word retransmissions and K is the number of information bits per word. Without loss of generality, we know that E_{rtx} is a function of n, of the ARQ strategy, and of E_{w}. For example, if the ARQ strategy is "Stop-and-Go" [7], there is barely no energy cost incurred by retransmissions, since the word to retransmit is already on the link.

We proceed by expressing the average transfer delay experienced by the last word in the FIFO. The computation of an average delay is too complex for the limited resources available to an on-chip link controller [10]. As a result, we approximate the average delay by the delay experienced by the last word queued in the FIFO. Let l be the number of words queued in the FIFO (i.e., the actual FIFO fill level) and denote by Ξ_{tot} the total number of cycles needed to transfer the last word of the FIFO. Assuming that the word errors are independent and identically distributed (i.i.d.), we can write

$$\mathbb{E}\left(\Xi_{\text{tot}}\right) = l \cdot \left(\Xi_{\text{w}} + \mathbb{E}\left(n\right)\Xi_{\text{rtx}}\right), \tag{2}$$

where $\mathbb{E}\left(\cdot\right)$ is the expectation operator and Ξ_{rtx} is the number of additional cycles incurred by a retransmission (dependent of the ARQ strategy). The average delay is approximated by $\frac{\mathbb{E}(\Xi_{\text{tot}})}{F_{\text{ch}}}$.

Finally, we define a link reliability metric, the *residual word error rate*, which is the probability that a word declared correct by the decoder is actually corrupted. We denote it by $\varepsilon_{\text{w}}^{\text{res}}$. Let ε_{b} the raw (i.e., the line) bit error rate. Obviously, ε_{b} is a (complex) function of the link parameters v_{ch} and F_{ch}. For i.i.d. bit errors, the residual word error rate is only a function of the raw bit error rate ε_{b} and of the error detecting code.

We now the express the link parameter control problem. Let $\overline{\Delta}_{\text{avg}}$ and $\overline{\varepsilon}_{\text{w}}^{\text{res}}$ be the constraints specified on the average transfer delay and residual word error rate. Theses two values are input by the processing element using the link. We state the link control problem below.

***Problem** 1:* Find the operating pair $(v_{\text{ch}}, F_{\text{ch}})$ which:

1) minimizes E_{b} as expressed in Eq. (1),
2) ensures that the performance constraint is met

$$\frac{\mathbb{E}\left(\Xi_{\text{tot}}\right)}{F_{\text{ch}}} \leq \overline{\Delta}_{\text{avg}}, \text{ and}$$

3) ensures that the reliability constraint is met $\varepsilon_{\text{w}}^{\text{res}} \leq \overline{\varepsilon}_{\text{w}}^{\text{res}}$.

This problem is highly complex, mainly due to the coupling of voltage and frequency: E_{b}, Ξ_{tot} and $\varepsilon_{\text{w}}^{\text{res}}$ are functions of v_{ch} and F_{ch}.

As shown in Problem 1, the link controller has to ensure that the residual word error rate does not exceed a desired value. However, it is impossible for the controller to receive feedback on the words delivered butcorrupted. Indeed, as shown in Fig. 3, the controller is only informed of *detected* word errors, and cannot distinguish

words transferred correctly from words affected by *undetected* errors. Even without any feedback on the occurrence of undetected errors, it would still be possible to guarantee a residual word error rate by (i) deriving an analytical model of the raw bit error rate and (ii) using it to forbid, a priori, operating pairs that do meet the reliability constraint. However, we reject such an approach that goes against the very principle of self-calibrating techniques. Moreover, there exists no accurate model of raw bit error rate and this fact is unlikely to change as the complexity of physical phenomena affecting functionality increases.

It follows from these observations that, in order to meet a reliability constraint, the link controller has to use the feedback it receives about detected word errors as a correlated feedback on undetected word errors. Again, because we do not wish to rely on a raw bit error rate model, the controller has to react by avoiding operating pairs that cause retransmissions. We state this important remark as a necessary assumption to solve Problem 1: *the link controller interprets retransmissions as a sign of unreliability*. This assumption is not needed if undetected errors cannot occur (as in [1]).

With the latter assumption, Problem 1 simplifies significantly. Since the link controller avoids operating pairs causing retransmissions, we can approximate the number of failed transfer attempts to 0. As a result, we rewrite Eqs. (1) and (2) as

$$\begin{aligned} E_\mathrm{b} &\approx \frac{E_\mathrm{w}}{K} \propto v_\mathrm{ch}^2 \text{ and} \\ \mathbb{E}\left(\Xi_\mathrm{tot}\right) &\approx \Xi_\mathrm{tot} = l \cdot \Xi_\mathrm{w} \propto l. \end{aligned}$$

With these relations, the solution of Problem 1 is straightforward, because the voltage and frequency are now decoupled. That is, E_b is a function of v_ch and not of F_ch, while the delay constraint depends on F_ch and not of v_ch. A self-calibrating link controller performs therefore the following:

Policy 1: To determine the operating pair $(v_\mathrm{ch}, F_\mathrm{ch})$,

(1) choose the slowest frequency F_ch that ensures $\frac{\Xi_\mathrm{tot}}{F_\mathrm{ch}} \leq \overline{\Delta}_\mathrm{avg}$, and
(2) for the frequency obtained in (1), determine the lowest voltage v_ch that does not cause detected word errors.

This policy corresponds to the layered control depicted in Fig. 2. The reliability constraint does not appear in Policy 1 because the maximum residual word error rate is determined by the encoding scheme only, and cannot be traded-off for energy. In case the reliability constraint is not met, the robustness of the encoding scheme has to be improved. We discuss reliability issues in Sec. III.

Item (1) of Policy 1 is a typical scheduling problem encountered in dynamic voltage scaling techniques. As such, it can be solved by a circuit that computes the delay resulting of each frequency (since we can assume there are only a limited choice of discrete frequencies). However, the choice of frequency gets complex when the average delay is not approximated (as we have presented). We do not develop this question here. On the contrary, item (2) of Policy 1 defines a more original problem. We call it the *self-calibrating control problem* and discuss it further in Sec. IV.

III. Self-calibrating communication

This section introduces a communication channel modelling errors occurring due to over-aggressive link operation. We call such errors *timing errors*. Then, we propose an encoding scheme that is specifically targeted to detect timing errors and present the resulting residual word error rate. Our study includes bit error rates as large as 1.

In order to perform item (2) of Policy 1, a self-calibrating controller has to periodically decrease the voltage used for a particular frequency and observe whether this causes retransmissions. When doing so, it may very well happen that the new frequency is too fast for the applied voltage. As a result, the received signal will be sampled even if many or all transitions failed. We call a timing error the sampling of a bit line before the completion of a transition. We assume that transition failures at a given time on a given line are independent from failures at other times, or on other lines. These assumptions lead to the communication channel depicted in Fig. 4. The failure of a transition is modelled by a Bernouilli source, and the raw bit error rate ε_b is the probability of the Bernouilli source output to be 1.

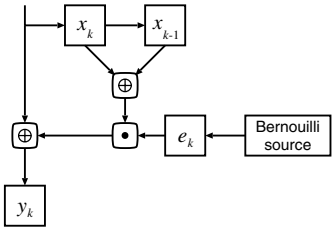

Fig. 4. Timing error channel. The dot operator denotes bitwise *and* of input values. The link value at time k is denoted by x_k.

Error detecting schemes deployed in self-calibrating links should exhibit a very low residual word error rate for large raw bit error rates. When the raw bit error rate reaches 1, the output of the timing error channel y_k is exactly its previous input x_{k-1} because all transitions deterministically fail. Classic error detecting codes, such as CRCs, only add spatial redundancy. As a result, they do not detect timing errors in this situation (i.e., $\varepsilon_\mathrm{w}^\mathrm{res} = 1$), since x_{k-1} is a codeword. Due to this fact, classic error detecting codes cannot be used in self-calibrating links. Alternating-phase codes are a simple extension of classic error detecting codes that incorporate temporal information about the data sequence within the redundant bits [8]. They detect all timing errors (i.e., $\varepsilon_\mathrm{w}^\mathrm{res} = 0$) when the raw bit error rate is 1. We contrast the residual word error rate of a classic 8-bit CRC code (CRC-8) with the one of a CRC-8 alternating-phase code in Fig. 5. We distinguish three raw bit error rate ranges. A first range covers raw bit error rates up to approximately 10^{-2}. In this region, the residual word error rate is acceptable and word errors occur relatively infrequently. This is fortunate, since, in this region, the link controller misinterprets them as a sign of unreliability. A second range

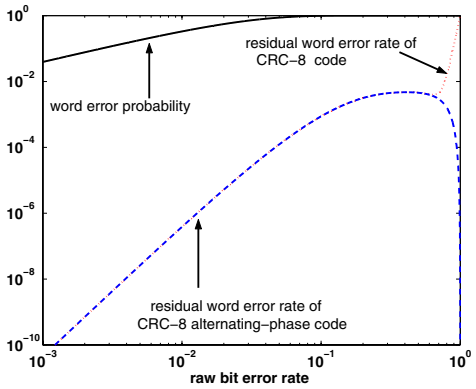

Fig. 5. The bottom dashed curve approximates the CRC-8 alternating-phase code residual word error rate, while the middle dotted curve plots the residual word error rate of the classic CRC-8 code. The top curve is the probability of a word error, i.e., the probability that at least one bit of the received word is corrupted.

covers moderate bit error rate (approximately 0.01 to 0.5). Here, the residual word error rate is relatively high and reaches a plateau. Even though word errors are frequent and correctly interpreted as a sign of unreliability, the link controller should avoid this region. Finally, a last range covers extremely high bit error rate (0.5 to 1.0) where word errors are quasi-certain. The residual word error rate of the classic CRC-8 code tends to 1, while the CRC-8 alternating-phase code detects an increasing number of errors. This region is periodically visited by the link controller, and excludes the use of classic error detecting codes such as the CRC-8.

Because, a priori, any bit error rate can be encountered, the residual word error rate guaranteed by the CRC-8 alternating-phase code is roughly of 10^{-2} (i.e., the maximum value of the curve). This maximum can be decreased by exploiting a latency-reliability trade-off [8]. Furthermore, we expect the middle range of bit error rate to correspond to a very tiny voltage range (for a fixed frequency), which results in the link controller to remain mostly in the first bit-error rate range and, sometimes, explore the third range. We have verified, with a bit error rate model we derived, that the transition to large bit error rate is indeed very steep: the bit error rate increases of approximately 4 orders of magnitude as the voltage decreases by 0.1V. A similar behavior has been observed for a multiplier [1].

IV. Self-calibrating control problem

In Sec. II, we have explained how the link control problem can be transformed into two independent problems. The first one (item (1) of Policy 1) regards the choice of the minimum link frequency meeting a performance constraint. We do not discuss this complex scheduling problem that has been studied in depth by researchers on dynamic voltage scaling techniques.

The second problem (item (2) of Policy 1) consists in tracking the minimum voltage that, for a given frequency, causes no detected errors. Solving this problem assumes the existence of:

- an encoding scheme on which the link reliability depends, and
- a control algorithm that adjusts a one-dimensional parameter (here, v_{ch}) in the presence or absence of detected word errors.

The latter issue is not critical in the sense that naive threshold-based algorithms grant significant energy saving, while not compromizing reliability [10]. Such an algorithm is depicted in Fig. 6. Some work is needed to study statistical properties (such as the average voltage tracking error) of threshold-based control algorithms. However, simple threshold-based algorithms are limited to the control of a one-dimensional parameter (in this case, the voltage). More sophisticated techniques are definitely needed to control more than one parameter. Problems of that kind are likely to appear in a near future. For example, researchers have already investigated the control of threshold voltage to limit static power dissipation [3], which suggests that self-calibration techniques could be extended to adjust supply and threshold voltages.

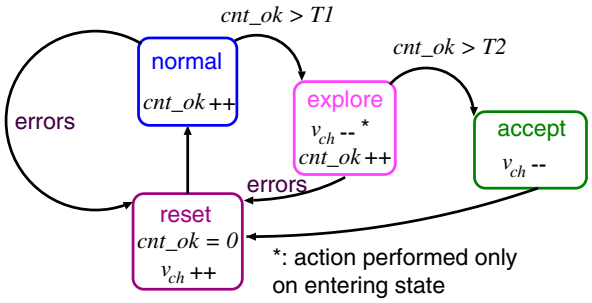

Fig. 6. The link controller counts the number of correct words transferred (*cnt_ok*). If errors occur, the voltage is increased and the counter reset. If the counter exceeds a given threshold (*T1*), the controller decreases the voltage (as indicated, only on entering state "explore"). The new voltage is accepted only if the counter exceeds another threshold (*T2*>*T1*). Otherwise, the previous voltage is restored. In both cases, the counter is reset.

The self-calibrating control problem is very general in its nature. It can be extended to processing elements. In doing so, the main difficulty is to obtain a feedback on the functional correctness of the processing element.

V. Conclusion

We have illustrated the concept of self-calibration with an on-chip link. A self-calibrating link consists of a classic link, augmented by a link controller, an ARQ controller, and an error detecting circuitry. We have shown that, even though the link bit error rate is sometimes extremely high, the error detecting circuitry can still ensure reliable operation. This study requires a communication channel model, but is independent of any bit error rate model. We have then stated the problem of controlling adaptively the link voltage and frequency. In a situation where undetected word errors are possible, we have shown that, since a self-calibrating link cannot rely on a bit error rate model, the control of voltage and frequency is decoupled (i.e., the implementation can be layered). In particular, we have identified and discussed a one-parameter (voltage) self-calibrating control problem that has a tangible application potential.

We argue that self-calibrating techniques can be seamlessly integrated in networks-on-chip. Regarding reuse cores, they bring complementary techniques to networks-on-chip. Furthermore, they enable to tolerate uncertainty on noise sources. This feature is of high interest for communication architectures automatically designed, as it may be the case in networks-on-chip.

References

[1] T. Austin, D. Blaauw, T. Mudge, and K. Flautner. Making typical silicon matter with Razor. *Computer*, 37(3):57–65, Mar. 2004.

[2] K. Flautner, S. Reinhardt, and T. Mudge. Automatic performance setting for dynamic voltage scaling. In *Proceedings of the 7th Conference on Mobile Computing and Networking*, pages 260–71, Rome, July 2001.

[3] S. M. Martin, K. Flautner, T. Mudge, and D. Blaauw. Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads. In *Proceedings of the International Conference on Computer Aided Design*, pages 721–25, San Jose, Calif., Nov. 2002.

[4] T. Pering, T. Burd, and R. Brodersen. The simulation and evaluation of dynamic voltage scaling algorithms. In *Proceedings of the International Symposium on Low Power Electronics and Design*, pages 76–81, Monterey, Calif., Aug. 1998.

[5] L. Shang, L.-S. Peh, and N. K. Jha. Dynamic voltage scaling with links for power optimization of interconnection networks. In *Proceedings of the 9th International Symposium on High-Performance Computer Architecture*, pages 91–102, Anaheim, Calif., Feb. 2003.

[6] P. P. Sotiriadis and A. P. Chandrakasan. A bus energy model for deep submicron technology. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, VLSI-10(3):341–50, June 2002.

[7] J. Walrand and P. Varaiya. *High-Performance Communication Networks*. Morgan Kaufmann, San Mateo, Calif., second edition, 2000.

[8] F. Worm, P. Ienne, and P. Thiran. Soft self-synchronising codes for self-calibrating communication. In *Proceedings of the International Conference on Computer Aided Design*, San Jose, Calif., Nov. 2004.

[9] F. Worm, P. Ienne, P. Thiran, and G. De Micheli. An adaptive low-power transmission scheme for on-chip networks. In *Proceedings of the 15th International Symposium on System Synthesis*, pages 92–100, Kyoto, Oct. 2002.

[10] F. Worm, P. Ienne, P. Thiran, and G. De Micheli. A robust self-calibrating transmission scheme for on-chip networks. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, VLSI-13(1):126–39, Jan. 2005.