Optimizing Checking-Logic for Reliability-Agnostic Control of Self-Calibrating Designs

Frédéric Worm, Patrick Thiran, Paolo Ienne
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne (EPFL)
{frederic.worm, patrick.thiran, paolo.ienne}@epfl.ch

***Abstract*— Self-calibrating designs have recently gained momentum as an alternative to methods relying on worst-case characterisation of silicon [2], [4], [8]. So far, reliable operation of existing link checkers—double sampling flip-flops or code-based—is not ensured over the whole range of bit error rate. Therefore, bit error rates where the checker reliability is poor are avoided either by worst-case characterisation of the link error rate (such as for double sampling flip-flops), or by constraining the operating point controller to avoid such regions (such as for code-based checkers).**

This paper proposes a novel checker architecture that bridges the gap between low overhead and high robustness over the whole error rate range. Specifically, we show how to combine optimally double sampling flip-flops with a code-based checker (in point of reliability). The resulting checker enables simple, efficient, and reliability-agnostic operating point control policies.

I. Introduction

There is a common agreement in the research community that a radical paradigm shift is needed in the way design uncertainties—such as noise, actual operating conditions, and silicon characteristics—are handled [1]. Currently, worst-case methods target error-free operation even in the most defavorable conditions, i.e., high noise level, high temperature, and the most pessimistic process corners. As CMOS technology scales down, the effect of various noise sources is exacerbated by the reduction in supply voltage and the emergence of new deep sub-micron effects. Simultaneously, variations in the features of silicon (e.g., performance and leakage current) increase dramatically. All these effects contribute to rendering worst-case methods highly complex—among others, due to huge verification costs—and overly pessimistic, which prevents designers from exploiting the full capabilities of typical silicon and thereby annihilates the efforts spent by technologists.

Researchers have recently proposed to apply digital self-calibration techniques to determine the operating-points of a circuit, instead of relying on worst-case characterisation [2], [8]. Fig. 1 depicts the high-level structure of a circuit with self-calibrating supply voltage. The controller adjusts the supply voltage depending on the occurrence of detected errors. While self-calibration requires some hardware overhead (mainly the checker and operating point controller logic), operating the circuit at sub-critical voltage offsets it.

It has been shown that circuits with self-calibrating operating-points are practical and attractive [2], [8]. In this paper, we focus on the development of *highly reliable* checkers with *minimal* wiring and circuitry overhead that do not restrict the operating-point controller (e.g., by the need of avoiding operating regions where undetected errors are likely to occur). Specifically, we propose the first checker architecture which is (i) low overhead, and (ii) highly robust to timing errors (i.e., errors that arise during the self-calibration process) under any possible error rate from 0 to 100%. We emphasise in Sec. II that no comparable checker exists to date. Then, the paper shows how a novel checker architecture is obtained by combining optimally double sampling flip-flops and a code-based checker. Interestingly, we show that, for such checkers, error correction capabilities are detrimental to reliability.

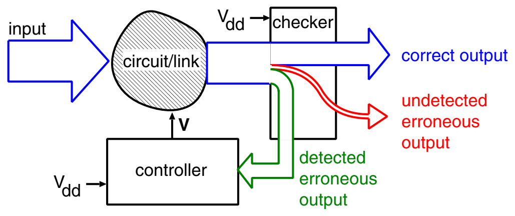

Fig. 1. A circuit with self-calibrating voltage supply. The output is verified by the checker informing the controller of detected errors. Voltages resulting in a high probability of undetected error should be avoided, which either requires worst-case characterisation or complexifies the voltage control policy.

Thanks to the novel checker architecture proposed in the paper, key design objectives can be decoupled: reliability remains confined to the checker, while the controller exploits power/performance trade-off. Currently, worst-case methods become increasingly complex due to their impossibility to decouple such design objectives.

A. Outline of the Paper

First, we introduce in Sec. II the two main checkers used in designs with self-calibrating operating points. By a qualitative discussion, we emphasise their complementarity in Sec. III and introduce a new reliability metric suited to checkers deployed in self-calibrating designs. Next, Sec. IV introduces a novel combination of double sampling flip-flops and code-based checkers that exploits optimally their complementary error detection capabilities. Finally, Sec. V shows simulation results confirming the superiority of the novel checker and Sec. VI summarises the achievements.

II. Existing Checkers

This sections introduces the two checker architectures existing to date for designs with self-calibrating operating points,

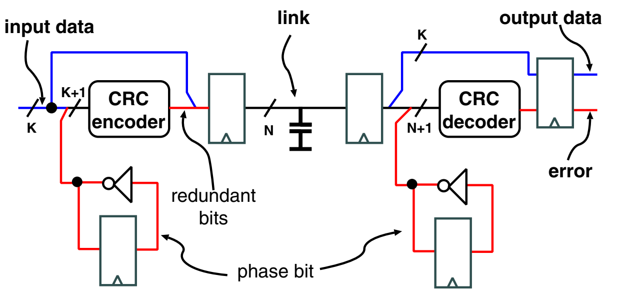

Fig. 2. CRC alternating-phase encoding. The K-bit information and the N-bit received data is augmented with the phase bit before computing the parity-checks.

namely soft self-synchronizing codes [7] and Razor flip-flops [2]. These two checkers feature high detection capabilities towards *timing* errors.

A. Soft Self-Synchronizing Codes

Soft self-synchronizing codes (abbreviated *soft SSC* throughout the paper) determine with high probability whether data sampled at the output of a link is correct—even though individual bit transitions may not have completed. A soft SSC is thus a checker specific to communication. Traditional codes add spatial redundancy in order to detect or correct additive errors. However, spatial redundancy cannot detect massive timing errors, whereby many individual bit transitions have not completed at the time the link output is sampled. Indeed, in a situation where all bit transitions would fail—which may happen during the self-calibration process—a coding scheme adding only spatial redundancy would sample the same codeword twice consecutively, resulting in an undetected error.

We describe the *alternating-phase* encoding, a soft SSC recently proposed [7]. The encoding includes a binary information about data sequencing into the spatial redundancy of a traditional error detecting code. This binary information, called *phase*, is the parity of the data index. The resulting encoding structure is depicted in Fig. 2. The phase bit is not transmitted but is generated locally by the encoder and decoder. In other words, the alternating-phase encoding alternates two dictionaries, C_0 and C_1 where C_0 (respectively C_1) is the set of codewords obtained by encoding information bits augmented with a 0 (respectively 1) phase bit. The sampled data is declared ready if and only if it belongs to the expected dictionary (i.e., C_0 or C_1 depending on the parity of the data index).

Alternating-phase encoding cannot correct errors, since the number of corrupted bits may easily be much larger than the correcting capability of any practical coding scheme. As a result, corrupted data is discarded and retransmitted. The residual error rate of alternating-phase encoding has been accurately approximated [7] and is shown in Fig. 3. We point out that the CRC alternating-phase encoding detects timing errors reliably both under small and large error rates; however, the checker does not operate reliably under moderate error rates where the phase of the received data is significantly mixed between correct (bit from current data piece) and wrong (bit from previous data piece).

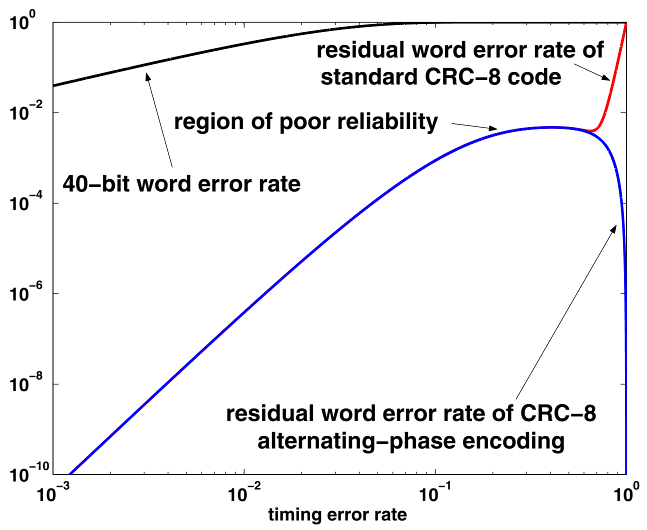

Fig. 3. Residual word error rate of the standard 8-bit CRC generated by the polynome $x^8 + x^2 + x + 1$, and of alternating-phase encoding using the same code for 32 information bits. The addition of the phase-bit makes it possible to detect reliably timing error under large error rate.

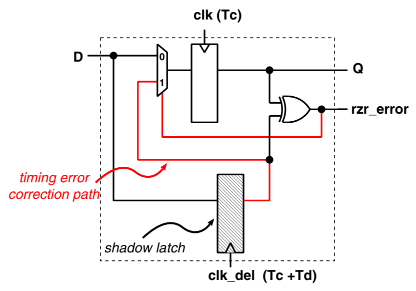

Fig. 4. A Razor flip-flop. The timing constraints of the shadow latch are guaranteed to be met (worst-case assumptions), so that its output is not only used to detect a timing error, but also to correct it.

B. Razor Flip-Flops

A Razor flip-flop [2] is drawn in Fig. 4. It consists of a main flip-flop—sampling with the normal clock (every T_c)—and a so-called *shadow-latch* fed by a clock delayed with respect to the main clock. Data at the input is latched a first time by the main flip-flop and a second time after some delay (T_d) by the shadow latch. The setup and hold constraints of the latter are ensured by both worst-case and best-case assumptions about the data arrival time. Provided these assumptions are met, the data held by the shadow latch acts as a "reference". A metastability-tolerant comparator validates the data latched in the main flip-flop by comparing it with the one of the shadow latch. In case a timing error—i.e., a mismatch—is detected, the output of the shadow latch is re-issued at the main flip-flop input, while at the same time raising an error signal. The timing error is thus corrected at the expense of a one cycle latency.

In order to operate reliably, the data at the flip-flop input should neither arrive too late nor to early. If it arrives so late that even the shadow latch samples a corrupted data piece, a timing error occurs and is not detected. On the contrary, if the next data piece arrives very early (less than T_d after the sampling by the main flip-flop), the output of the shadow latch may invalidate the correct output of the main flip-flop. If so, an error is reported although none occurred and reliability is infringed since the shadow latch corrects with a wrong data piece. This error process is called a *short-path* error.

Contrary to soft SSC that are specific to communication, Razor flip-flops are a generic checker since they can replace standard flip-flops after any combinational logic block as long as best and worst case assumptions required for their reliable operation are met. In the remaining of the paper, we focus on checkers for a self-calibrating on-chip link.

Our contribution is to combine optimally alternating-phase encoding with double sampling flip-flops in order to detect timing errors reliably and cheaply over the whole error rate range, which is achieved neither by Razor flip-flops nor by soft SSC used individually.

III. Qualitative Discussion

In this section, we emphasise the complementarity of Razor flip-flops and soft SSC by a qualitative discussion. The conclusions drawn from the comparison are confirmed later by the simulations presented in Sec. V. Moreover, we introduce a reliability metric specific to checkers used in a self-calibrating design.

We are interested in the *residual* and *reported* error probability as a function of the link supply voltage. This information characterises completely the quality of a checker. As with any checker, the data delivered to the end-user should not be corrupted by residual errors. In addition, the checker informs the controller of each detected error, so that unsafe operating points—supply voltage, in the particular case—can be dynamically avoided. Let t_p be the propagation delay through a bit line. We consider t_p a random quantity parametrised by the supply voltage v_c: for each particular value of v_c, t_p is characterised by a probability distribution. We perform a qualitative comparison because the points we want to make depend on important features of the checkers and not on a particular bit error rate model (i.e., on the actual relation between t_p and v_c). By definition, a bit line is affected by a timing error whenever $t_\mathrm{p} \geq T_\mathrm{c}$ with T_c the sampling period.

A single Razor flip-flop is affected by a residual bit error if and only if $t_\mathrm{p} \notin [T_\mathrm{d}; T_\mathrm{c} + T_\mathrm{d}]$ with T_d the delay between the clock fed to the main flip-flop and the clock fed to the shadow latch. As mentioned in Sec. II-B, short-path errors compromise reliability and occur whenever $t_\mathrm{p} \leq T_\mathrm{d}$. Undetected timing errors happen whenever $t_\mathrm{p} \geq T_\mathrm{c} + T_\mathrm{d}$, i.e., when both the main flip-flop and shadow latch sample the same corrupted data. As a result, reliable operation of a Razor flip-flop occurs in limited range of voltages, where the probability of a short-path and an undetected timing error are both acceptably low. In practice, this is ensured by introducing buffers delaying data arrival—avoiding thus short-path errors—while undetected timing errors are avoided thanks to worst-case characterisation of t_p.

Considering a group of Razor flip-flops (such as in a Razorised bus), a word error is reported when at least one of the Razor flip-flops reports an error. Equivalently, the error signal fed to the controller is obtained by *OR*ing the individual error signals of each Razor flip-flop. A residual word error

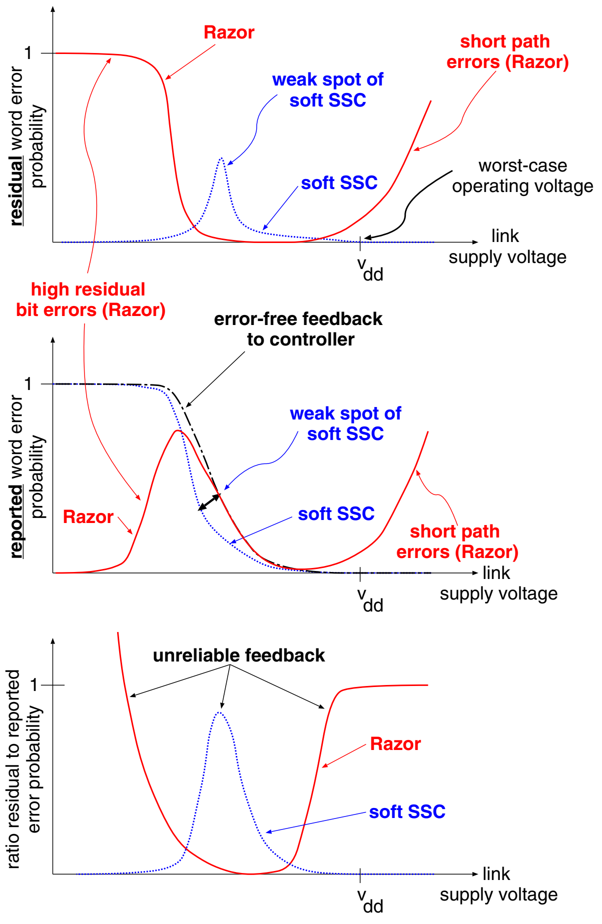

Fig. 5. Top and middle: residual (top) and reported (middle) word error probability as a function of supply voltage for a bus terminated by Razor flip-flops or a soft SSC. Bottom: ratio of residual to reported error probability for each checker.

occurs whenever at least one Razor flip-flop is affected by a residual bit error, even though other Razor flip-flops than the one(s) causing a residual error may report an error. In a situation where at least one Razor flip-flop is affected by a residual bit error and at least one Razor flip-flop reports an error, residual and reported word errors are not exclusive events.

The top (respectively middle) graph of Fig. 5 compares the residual (respectively reported) word error probability of the two existing link checkers: a group of Razor flip-flops and a soft SSC. The latter operates satisfactorily both under small and large error rates—as expected from Fig. 3—where Razor flip-flops suffer from short-path or undetected timing errors. On the contrary, Razor flip-flops provide correct information to both the end-user and the controller in the operating area where the soft SSC is unreliable.

Now, we introduce the reliability metric used in the paper. A checker used in a self-calibrating design should provide reliable information about both *residual* errors (feedback relevant for the end-user) and *reported* errors (feedback relevant for controller). The particularity of Razor-based checkers is that reported and residual errors are not exclusive. Thus, there is no simple relation between these two quantities, contrary to soft

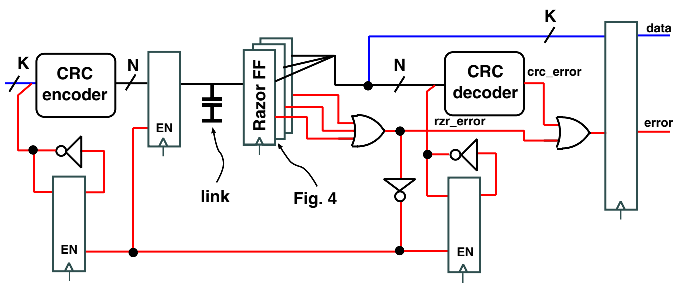

Fig. 6. Checker combining Razor flip-flops and a soft SSC as a minimal extension of Fig. 2. Each line of the link is sampled by a Razor flip-flop before the decoder validates the output of all Razor flip-flops.

SSC where any undetected error is necessarily residual and vice-versa. As a result, we measure the reliability of a checker as the ratio of the residual to the reported error probability and call the resulting metric $\rho = \varepsilon_{\mathrm{w}}^{\mathrm{res}}/\varepsilon_{\mathrm{w}}^{\mathrm{det}}$. The smaller the metric is, the more reliable the checker. The complementarity of Razor flip-flops and soft SSC is obvious by comparing their respective reliability ratio, which is sketched in the bottom graph of Fig. 5. This graph conveys compactly the information expressed in the top and middle graphs of the figure.

The next section introduces two checker architectures combining double sampling flip-flops and soft SSC.

IV. The Combined Checkers

The checker combining Razor flip-flops and a soft SSC is described in Fig. 6 as a 3-stage pipeline. In the text, we denote this checker *RZR+CRC*. The binary signal *crc_error* indicates an error in the decoding. The binary signal *rzr_error* indicates the detection of at least one timing error among the N Razor flip-flops sampling the link output. The decoding occurs after the Razor flip-flops, allowing thus to combine error signals from the decoder and the Razor flip-flops very easily: the error signal output along with the decoded data is the *OR* of *crc_error* and *rzr_error*.

As shown in Fig. 6, the phase of the decoder needs to be kept unchanged whenever a timing error is detected because the data corrected by Razor flip-flops is output one cycle after the error is reported. Freezing the phase of the decoder during a cycle also requires to freeze the encoder phase during the same cycle, since the two phases must stay synchronised. This imposes a severe timing constraint on the backward line propagating to the encoder flip-flop enable port.

The checker architecture of Fig. 6 has already been proposed as a significant improvement on individually Razor flip-flops or soft SSC [9]. In this paper, we show that the error correction capabilities of Razor flip-flops are detrimental to reliability under large error rates. One of the reason is very intuitive: the output of the shadow latch is always used as a reference to correct timing errors, even when it is itself corrupted. Therefore, we propose a novel checker architecture depicted in Fig. 7 whereby the shadow latch is replaced by a shadow flip-flop used in the sole purpose of error detection. While the RZR+CRC checker combines each individual checker serially—the soft SSC validating the razorised data—the architecture of Fig. 7 is a parallel combination, since

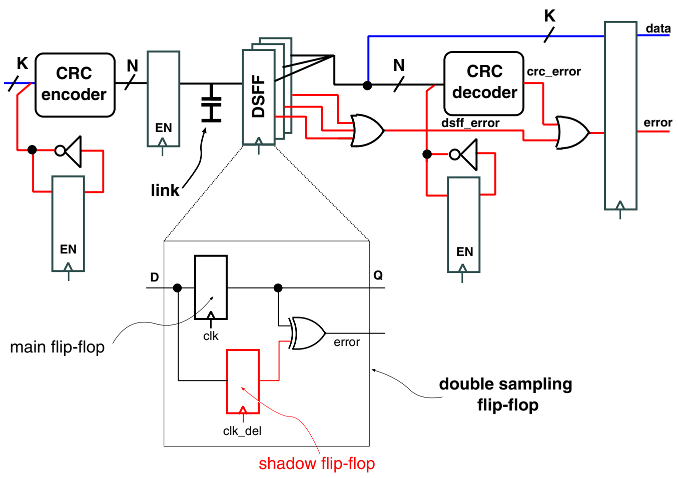

Fig. 7. Checker combining double sampling flip-flops and a soft SSC. The double sampling flip-flops are used only in the purpose of detecting timing errors. Contrary to Fig. 6, there is no need to synchronise the encoder and decoder phase when timing errors are detected.

the decoder of the soft SSC is fed with exactly the same data as the double sampling flip-flops. We call this checker architecture *DSFF+CRC*. It can easily be shown that *any* word error detected by either Razor flip-flops, or a soft SSC, or the RZR+CRC checker is also detected by the DSFF+CRC checker. Clearly, any error detected by a Razor flip-flop is detected as well by the DSFF+CRC checker since the same delay is used between the main and shadow flip-flops. Moreover, any error detected by the soft SSC is also detected by the DSFF+CRC checker since the decoder of both checkers are fed with exactly the same data pieces. At last, the decoder of the RZR+CRC checker is fed with a data piece differing from the one fed to the decoder of the soft SSC checker only when at least one Razor flip-flop reports an error—in which case, the DSFF+CRC checker also detects the error. Otherwise, the decoder of the RZR+CRC and DSFF+CRC checkers are fed the same data and detect thus the same errors. As far as residual errors are concerned, a word error is undetected by the DSFF+CRC checker if and only if any timing error occurring is undetected by the double sampling flip-flops terminating the bit line *and* the resulting word error is undetected by the soft SSC. We will observe in the next section that such a combination of events is highly unlikely.

For both architectures of Figures 6 and 7, an entity called ARQ (Automatic Repeat reQuest) is required to schedule retransmissions and ensure in-order data delivery. The DSFF+CRC checker retransmits a data piece whenever an error is reported by at least one double-sampling flip-flop *or* by the decoder. On the contrary, the RZR+CRC checker schedules a retransmission only when the decoder reports an error. We have synthetised an ARQ controller for the RZR+CRC checker—in fact, the most complex configuration—and found that it consists of only 6 flip-flops and about 20 gates.

A. Hardware Complexity

The novel checker proposed in the paper combines double sampling flip-flops with soft SSC. We argue that the incurred hardware overhead is minimal. First, the area overhead caused by double sampling flip-flops can be mitigated by using scan flip-flops in the purpose of double sampling [6]. Second, we show in Sec. V that a single alternating-parity bit protecting 8 information bits offers a high robustness to timing errors over the whole range of bit error rates. Such a checker only adds a single redundant wire and a minimal codec circuitry needed to compute a parity. Moreover, the amount of redundancy ($1/8$ or 12.5%) is smaller than or comparable to the one added by error control techniques traditionally deployed over on-chip bus—e.g., [5]. The codec circuitry is also less complex as the DSFF+CRCR checker does not perform error correction.

Finally, the hardware overhead of the RZR+CRC or single soft SSC checker is larger than the one incurred by DSFF+CRC, because more redundant bits are required for the same reliability level.

V. Robustness To Timing Errors

We have contrasted qualitatively the detection capabilities of Razor flip-flops and soft SSC in Sec. III. We compare now thoroughly the reliability of the combined checkers (i.e., RZR+CRC and DSFF+CRC) with Razor flip-flops and a soft SSC.

A. Experimental Set-Up

We have simulated in VHDL a system consisting of a FIFO, an encoder, a bus terminated by Razor or double sampling flip-flops, a decoder, and an ARQ controller. Each simulation has been run until 500,000 eight-bit data pieces randomly generated were transferred in sequence. For large word error rate, the actual number of word transfers is actually much larger due to frequent retransmissions (up to 10^7 transfers).

We explain now the simulation set-up. We have modelled how the delay through a bit line of the link t_p depends on the supply voltage v_c by using the lumped capacitance wire model:

$$t_\mathrm{p} = \frac{C_\mathrm{L}}{k_\mathrm{m}} \cdot \frac{v_\mathrm{c}}{\left(v_\mathrm{c} - v_\mathrm{th}\right)^2}, \tag{1}$$

with k_m the transistor transconductance, C_L the line capacitance, and v_th the device threshold voltage. We refrain from using more complex delay models for two reasons: first, the lumped capacitance model is widely used in the design tools. Second, we study the reliability of the checkers for any error rate from 0 to 100%. In that sense, the presented results are independent of the actual relation between the link supply voltage and the bit error rate.

To model the variability of t_p, we describe the ratio $C_\mathrm{L}/k_\mathrm{m}$, which we denote by α, by a Gaussian random variable: $\alpha \sim N\left(\mu_a, \sigma_a\right)$. The coefficient $v_\mathrm{c}/\left(v_\mathrm{c} - v_\mathrm{th}\right)^2$ in the right hand side of Eq. (1) is considered as deterministic. Accordingly, t_p is also a Gaussian random variable: $t_\mathrm{p} \sim N\left(\mu_{t_\mathrm{p}}, \sigma_{t_\mathrm{p}}\right)$ where

$$\mu_{t_\mathrm{p}} = \mu_a \frac{v_\mathrm{c}}{\left(v_\mathrm{c} - v_\mathrm{th}\right)^2} \quad \text{and} \quad \sigma_{t_\mathrm{p}} = \sigma_a \frac{v_\mathrm{c}}{\left(v_\mathrm{c} - v_\mathrm{th}\right)^2}.$$

The bus is modelled in non-synthetisable VHDL and introduces random timing errors. That is, during simulated word transfers, we generate for each bit line a random value for the delay t_p through the line according to a Gaussian distribution parametrised by the supply voltage v_c. Henceforth, we determine for each line whether a timing error occurs (i.e., when $t_\mathrm{p} \geq T_\mathrm{c}$) and, if that is the case, whether the error is detected by the Razor or double sampling flip-flop (i.e., when $t_\mathrm{p} \leq T_\mathrm{c} + T_\mathrm{d}$). A Razor flip-flop affected by an undetected timing error outputs the previously sampled data, even during a correction cycle triggered by (at least) one other Razor flip-flop that has detected an error. The delays t_p through different bit lines are modelled as independent and identically distributed. As the voltage is lowered, the bit error rate rises identically on all lines but each line remains statistically independent from the others.

We model a link manufactured in a 130nm CMOS technology with the following parameters: the nominal voltage v_dd is 1.2V, the threshold voltage v_th is 0.2V, and the cycle time T_c amounts to 2ns. The value of the parameter μ_{t_p} (i.e., the typical delay) is set to 1ns under the nominal voltage v_dd. Finally, the value of the parameter σ_{t_p} is set so that the probability of a timing error under the nominal voltage v_dd amounts to 10^{-15}. In addition, $T_\mathrm{d} = 0.3 T_\mathrm{c}$, as it has been reported for busses [3].

During each simulation, we point out that both voltage and frequency are fixed. However, the voltage is varied from 0.6V to 1.3V with 50mV increments (one simulation performed for each voltage value).

We compare five different checkers: *RZR*, i.e., 8 Razor flip-flops, *CRC3*, i.e., the 3-bit CRC generated by the polynome $x^3 + x + 1$, *RZR+CRC1*, i.e., a minimal combined checker augmenting RZR with an alternating-phase parity check, *RZR+CRC3* which combines RZR and CRC3, and finally *DSFF+CRC1* that combines 8 double sampling flip-flops with an alternating-phase parity check. We compute the reliability metric introduced in Sec. III by dividing the total number of residual errors with the total number of reported errors.

B. Comparison Results

Fig. 8 plots the reliability metric ρ of these five checkers and the word error rate as a function of the timing bit error rate. As expected, the combined checkers always outperform RZR. The RZR checker becomes highly unreliable as the bit error rate increases towards 100%, because the output of the shadow latch is used for timing error correction even though it is corrupted.

Comparing the RZR+CRC checkers with CRC3 reveals that the former operates reliably in most of the bit error rate range where the reliability of the single soft SSC checker is poor. In this region, the Razor flip-flops are able to correct most of the timing errors before submitting their data the decoder. Yet, as the bit error rate further increases, the CRC3 is now able to detect all timing errors since the phase of most bits input to the decoder conflicts with the one of the phase bit. On the contrary, the RZR+CRC checkers reach their maximum unreliability. The reason is that a large part of the Razor flip-flops cannot correct anymore timing errors reliably. As a result, the phase of the data input to the decoder is significantly mixed between correct and wrong, which is detrimental to the decoder detection capabilities (refer to Fig. 3). Nevertheless, comparing the CRC3 with RZR+CRC checkers in Fig. 8 shows that the region of poor reliability has shrunk and has

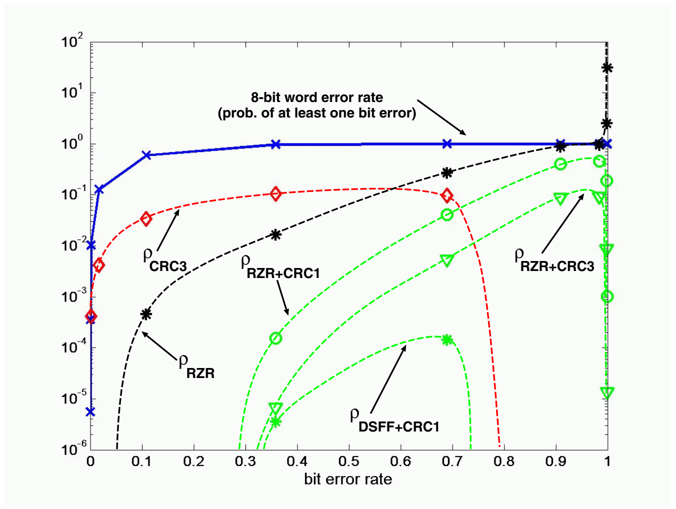

Fig. 8. The ratio $\rho = \varepsilon_{\mathrm{W}}^{\mathrm{res}}/\varepsilon_{\mathrm{W}}^{\mathrm{det}}$ as a function of the bit error rate. The curves are extrapolated from pointwise measurements. The DSFF+CRC1 checker necessarily outperforms the CRC1 and RZR+CRC1 checkers. The simulations show that it also outperforms the CRC3 and RZR+CRC3 checkers over the whole range of error rate.

been shifted to larger bit error rate values. It follows that the RZR+CRC checkers offer a larger range of bit error rate available to inform reliably the operating point controller that the link is not operative (note that, when the RZR+CRC checker becomes unreliable, the word error rate has already reached 100%). It is also worth pointing out that reducing the amount of spatial redundancy in the RZR+CRC checker—which decreases the wiring and circuitry overhead—does not degrade significantly reliability, especially under large error rate where mostly timing redundancy (such as the phase) matters.

Finally, comparing the RZR+CRC1/3 checkers with the DSFF+CRC1 checker in Fig. 8 confirms that giving-up to correction capabilities boosts significantly reliability. The gain is especially striking for bit error rates larger than approximately 0.7 where the DSFF+CRC1—like the CRC3—detects all timing errors. Because correcting timing errors decreases the error detection probability of the soft SSC under a large bit error rate, suppressing correction makes it possible to benefit from the high robustness of the soft SSC under large error rate. The price paid is a less efficient error recovery under small error rates where correction capabilities of Razor flip-flops are not detrimental to reliability.

The DSFF+CRC1 checker outperforms all other checkers despite its minimal wiring and circuitry overhead. In that sense, we claim that such a checker combines *optimally* double-sampling flip-flops with soft SSC, because the detection capabilities of each of its counterparts are used without any negative interference—contrary to the RZR+CRC checker. As Fig. 8 attests, the DSFF+CRC1 provides a reliable feedback to both the end-user and the operating point controller over the *whole* error rate range. Thereby, the controller is not restricted by the need of avoiding weak spots of the checker.

A possible variation on the DSFF+CRC checker consists in disabling dynamically correction capabilities of Razor flip-flops when the bit error rate is estimated large. The motivation stems from the curves of Fig. 8 plotting the reliability ratio of the RZR+CRC checker: correcting timing errors is efficient and does not infringe reliability unless the error rate grows large. As a result, a checker where the correction path through the shadow latch/flip-flop is disabled as soon as more than a certain number of timing errors (e.g., only one) are reported seems attractive. Here, hardware complexity (required to detect when a certain number of bits in a register are set) is traded-off for efficiency in error recovery.

The DSFF+CRC checker brings additional benefits besides its unique robustness to timing errors. Because the output of the shadow flip-flop is not used for correction (i.e., as a reference), short-path errors only cause false alarms and do not jeopardise reliability—contrary to the RZR and RZR+CRC checkers. Finally, double sampling provides also some robustness towards soft errors [6], which cannot be exploited safely by the RZR and RZR+CRC checkers (since the shadow latch does not necessarily hold the correct data piece).

VI. Conclusions

This paper has proposed a novel checker architecture for on-chip links with self-calibrating operating points. We have exploited the complementarity of double sampling flip-flops and a code-based checker and shown how to combine these two checkers optimally from a reliability point of view. Despite its minimal wiring and circuitry overhead, the resulting checker features a strong and unmatched robustness to timing errors over the whole range of bit error rate. Contrary to Razor flip-flops, reliability is ensured without relying on worst-case characterisation of the link. As a result, efficient, simple, and unrestricted (by weak spots in the checker) operating point control policies can be developed to fully exploit power/performance trade-off of self-calibrating circuits.

References

[1] T. Austin, V. Bertacco, D. Blaauw, and T. Mudge. Opportunities and challenges for better than worst-case design. In *Proceedings of the Asia and South Pacific Design Automation Conference*, January 2005.

[2] S. Das, D. Roberts, S. Lee, S. Pant, B. David, T. Austin, K. Flautner, and T. Mudge. A self-tuning DVS processor using delay-error detection and correction. *IEEE Journal of Solid-State Circuits*, 41(4):792–804, Apr. 2006.

[3] H. Kaul, D. Sylvester, D. Blaauw, T. Mudge, and T. Austin. DVS for on-chip bus designs based on timing error correction. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, pages 80–85, Munich, Mar. 2005.

[4] C. H. Kim, S. Hsu, R. Krishnamurthy, S. Borkar, and K. Roy. Self calibrating circuit design for variation tolerant VLSI systems. In *Proceedings of the 11th IEEE International On-Line Testing Symposium*, pages 100–5, Saint Raphaël, France, July 2005.

[5] C. McNairy and D. Soltis. Itanium 2 processor microarchitecture. *IEEE Micro*, 23(2):44–55, Mar.–Apr. 2003.

[6] S. Mitra, N. Seifert, M. Zhang, Q. Shi, and K. S. Kim. Robust system design with built-in soft-error resilience. *IEEE Computer*, 38(2):43–52, 2005.

[7] 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.

[8] 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.

[9] F. Worm, P. Thiran, and P. Ienne. Designing robust checkers in the presence of massive timing errors. In *Proceedings of the 12th IEEE International On-Line Testing Symposium*, pages 281–86, Lake of Como, Italy, July 2006.