Low power error resilient encoding for on-chip data buses

Davide Bertozzi
DEIS-University of Bologna
40136 Bologna Italy
dbertozzi@deis.unibo.it

Luca Benini
DEIS-University of Bologna
40136 Bologna Italy
lbenini@deis.unibo.it

Giovanni De Micheli
CSL-Stanford University
Stanford, CA 94305
nanni@stanford.edu

Abstract

As technology scales toward deep submicron, on-chip interconnects are becoming more and more sensitive to noise sources such as power supply noise, crosstalk, radiation induced effects, etc. Transient delay and logic faults are likely to reduce the reliability of data transfers across datapath bus lines. This paper investigates how to deal with these errors in an energy efficient way. We could opt for error correction, which exhibits larger decoding overhead, or for the retransmission of the incorrectly received data word. Provided the timing penalty associated with this latter technique can be tolerated, we show that retransmission strategies are more effective than correction ones from an energy viewpoint, both for the larger detection capability and for the minor decoding complexity. The analysis was performed by implementing several variants of a Hamming code in the VHDL model of a processor based on the Sparc V8 architecture, and exploiting the characteristics of AMBA bus slave response cycles to carry out retransmissions in a way fully compliant with this standard on-chip bus specification.

1. Introduction

The trend for digital integrated circuits exhibits shrinking of geometries, scaling of supply voltages, increased interconnect density, faster clock rates and higher integration levels. This process exposes technology to an increased sensitivity to a great number of noise mechanisms, such as capacitive and inductive cross-talk, power supply noise, leakage noise, charge sharing, process variations, soft errors, etc. [13] [2] [7] [1]. In this context, on-chip interconnections play a crucial role for the performance and the reliability of future ICs [4].

The impact of noise induced in an on-chip interconnect depends on the ratio of the induced noise over the receiving gate voltage noise margin. But as the supply voltage of ICs scales, these noise margins decrease, raising concerns on the reliability of data transfers across wires [6].

This problem is closely related to another major issue of ICs design: power consumption. Wires have been shown to account for a remarkable percentage of the total on-chip power dissipation (up to 40 or 50%) [9].

A straigthforward solution is the reduction of the voltage swing of signals propagated across interconnections. This requires the design of low-swing interfaces, that can also be combined with the use of low supply voltages to achieve a larger power reduction [17] [14].

All these trends force the designer to a tradeoff between power consumption and communication reliability. The lower the power supply and/or the interconnect swing, the more the sensitivity to noise sources because of the decreased noise margins.

2. Error detecting/correcting codes

For datapath bit lines, the reliability issue can be addressed making use of error detecting or correcting codes (EDC/ECC). Traditional codes used to implement VLSI self-checking circuits are based on error detection: parity code, two-rail code and all-unidirectional EDCs (such as m-out-of-n and Berger ones) [12]. As we move to consider faults affecting bus lines in deep sub-micron circuits, we have to deal with noise sources that may degrade the error detection effectiveness of these codes.

For example, the efficient Berger code is able to detect all unidirectional errors, but in [10] is shown that crosstalk may cause bidirectional and hence undetectable errors. The workaround proposed in [11], relying on weight-based codes, provides a Berger code with the ability to detect many crosstalk-related bidirectional errors, but requires the knowledge of the layout.

To better account for the nature of noise sources affecting bus lines, in this paper we focus on a Hamming code [8]. Its capability to handle all single and double errors and many multiple ones makes it attractive. Beyond its low-complexity codec, it is also very flexible, because it can be implemented either as a purely detecting code, or as purely

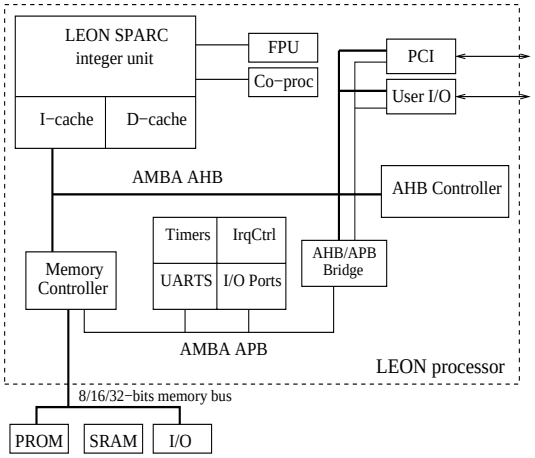

Figure 1. Leon processor block diagram

correcting one or with a combined approach of detection and correction.

Each version has different characteristics in terms of detecting and correcting capabilities. This allows us to investigate the most efficient course of action to take when a bus transfer is affected by errors. A Hamming code corrects only the error patterns of single error. In this case the recovery action (error correction) implies a relatively complex decoder but a very small timing penalty. On the contrary, if a Hamming code is used for error detection, a data word retransmission has to be triggered on the bus when an erroneous transfer is detected. This approach relies on simpler decoders and a better detection capability, but generates a larger number of bus transitions and degrades the performance because of the associated delay.

The comparison between the two possible recovery policies from erroneous transfers is addressed here from a power consumption viewpoint. We indicate which technique ensures a predefined level of noise tolerance at the minimum power dissipation, referring the results also to the original unencoded bus and to a traditional 1 bit parity code.

In order to perform this analysis, we focused on a case study. We considered a subset of the VHDL model of a 32 bit processor compliant to SPARC V8 architecture (called LEON [16]), and provided it with noise tolerance. As the considered system makes use of the AMBA bus specification for on-chip communication, we implemented retransmissions of incorrectly received data words in a way fully compliant with the AMBA bus standard.

In section III we describe the VHDL models we used in our simulations. In section IV the retransmission implementation for AMBA bus is presented. Energy efficiency of the considered error control schemes is investigated all over sections V, VI and VII. Simulation results are reported in section VIII, while in section IX conclusions are drawn.

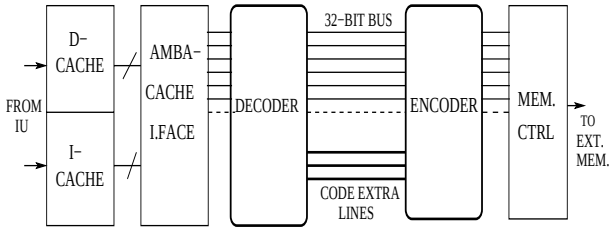

Figure 2. Noise tolerance implementation

3. Noise tolerant schemes

The Leon processor has a full implementation of the AMBA bus, consisting of a high performance system bus (AHB) and of a slower peripheral pus (APB), communicating with each other via a bridge (Fig. 1). The former is meant for high speed data transfers, and has the processor as the default master and two slaves (memory controller and APB bridge), in its basic configuration.

We focused on the read data bus (Fig. 2), accessed by the I- and D-cache whenever an instruction/data miss occurs. The cache accesses the bus by means of a hardware interface that generates the AMBA bus control signals and respects its timing requirements. The request for data is transmitted to the memory controller, that in turn accesses the off-chip memory and forwards read data back to the cache.

We have enhanced a read data path with noise tolerance by inserting an error control coder at the memory controller side, a decoder at the cache side and some additional bus lines needed to accommodate the different error detection/correction strategies we consider. In particular, we investigate the error detection capability and the energy efficiency of the following schemes:

Single Error Correction Bus - SEC: This is a basic implementation of a (38,32) Hamming code, with a single error correction capability. The purpose of this scheme is to highlight the characteristics of a recovery strategy based on correction. The decoder is more complex than the encoder, because of the correction circuitry. The bus requires 6 additional check bits.

SEC and Double Error Detection Bus - SECDED: A distance-3 Hamming code, like that implemented in SEC Leon, can easily be modified to increase its minimum distance to 4, adding one more check bit, chosen so that the parity of all bits, including the new one, is even [15]. This version of the Hamming code, that features 7 check bits instead of the 6 of the previous version, is usually used for single error correction and double error detection. Yet it allows to detect also all error patterns of an even number of errors, event though the double ones are the most

meaningful for their higher probability. In case a double or multiple error is detected, the recovery action would be the retransmission of the wrong data word. Here the codec is slightly more complex than that of the previous scheme because of the combined approach of correction and retransmission.

Error Detection Bus - ED: Using the Hamming code for detection purposes only, it is possible to exploit its full detection capability, which includes not only all single and double errors, but also a large quantity of multiple errors. An (n, k) linear code is capable of detecting $2^n - 2^k$ error patterns of length n. The undetectable error patterns are $2^k - 1$, and they are identical to the nonzero code words. This scheme exhibits the same encoder as SEC but a very simplified decoder, because it only has to compute and check the syndrome bits. This Leon implementation highlights the characteristics of a retransmission oriented approach.

Singol Parity Bit Bus - PAR: It is the simplest error detection scheme, that captures all errors in an odd number of bits. We consider this case, which involves a minimum overhead of only one extra parity bit line, for comparison with the other Hamming code implementations. The original version of Leon, without noise tolerance, will be considered as well, and will be referred to as "**ORIG".**

4. Retransmissions with AMBA bus

The approach to deal with the noise tolerance issue of ICs by means of checking schemes combined with a retry procedure has been widely discussed in the open literature [3]. Yet, emphasis is always given to the concurrent checking mechanism and not to the course of recovery action to take.

On the contrary, for on-chip data buses the issue of how to implement retransmissions of erroneous data words, given a decoder with detection-only capabilities, has to be addressed very early in the design stage, as it heavily impacts bus architecture and performance. This problem resembles the tradeoff between forward error correction (FEC) and automatic repeat request (ARQ), well-known to the communication community [18], but as regards on-chip buses its investigation is still in the early stage.

In this context, we decided to carry out data word retransmissions in a way fully compliant to existing bus standards. In particular, retransmissions for the AMBA bus specification have been implemented by just exploiting its characteristics.

In the AMBA bus standard, a slave can indicate that the transfer in progress cannot be completed successfully. In this case, only higher priority masters will gain access to the bus, and the present master should continue to retry the transfer until it completes. This mechanism is referred to as the slave $retry$ response, and is described in Fig. 3. In the penultimate cycle of a read transfer, the slave drives $HRESP$ to indicate $RETRY$, while driving $READY$ low to extend the transfer for an extra cycle. This signals configuration is sampled by the rising edge of the clock, and it causes the next cycle to be $IDLE$. During the idle cycle, $HRESP$ remains high, and this prevents the data on the bus to be sampled both by the cache and by the integer unit of the Leon processor on the next rising edge of the clock. It also causes the processor to retry the transfer at the same memory location, provided the control of the bus is not lost.

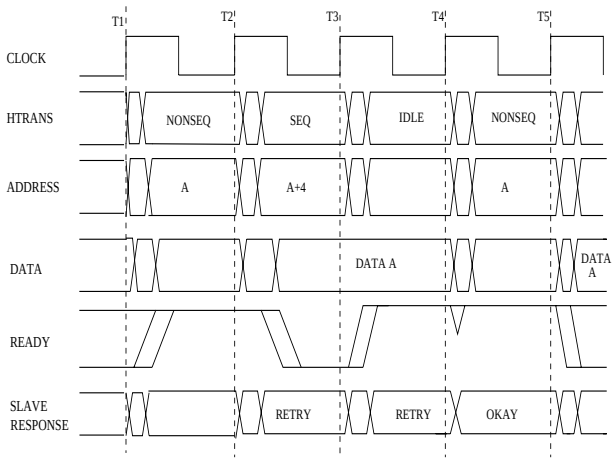

Figure 3. Waveforms of the retry response on an AMBA bus, exploited for retransmissions

This mechanism has been used to implement retransmissions of incorrectly received data. The assumption we make is that before the rising edge $T4$ of the clock, which should sample data as the $READY$ signal is high, the decoder evaluation has completed. This can be accounted by adding its delay (as well as the encoder delay) to the memory access time, and can result in a few additional wait states. If the decoder response indicates errors, the master takes the same actions that are triggered by the $HRESP$ signal in the standard way of working. That is, the internal signal allowing data sampling on $T4$ at the cache and at the integer unit is not asserted, while another signal is activated, that forces the processor to repeat the last transfer. The activation time of these signals should be considered in the computation of the needed wait states as well.

In this implementation, the $HRESP$ signal is not used at all, and the memory controller just serves two successive access requests to the same memory location. The entire mechanism is master-controlled.

Note that because of the pipelined nature of the AMBA

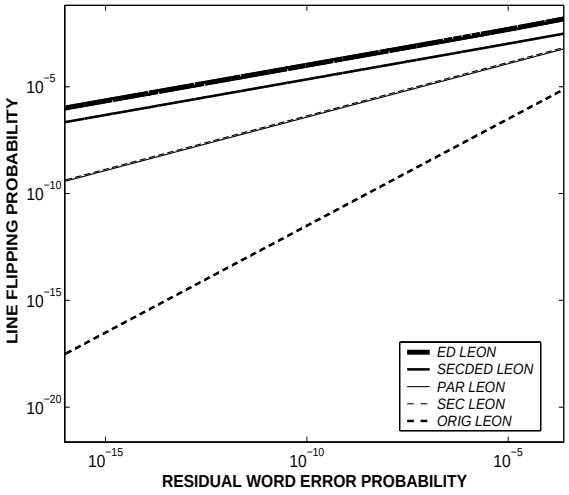

Figure 4. Requirements on line flipping probability to ensure a predefined level of reliability

bus, by the time the master issues a retry procedure, then the address for the following transfer has already been broadcasted onto the bus (see address $A + 4$ in Fig. 3). This involves additional transitions on the address bus to restore the address of the data word to be retransmitted. We will consider their impact over power consumption.

Provided the timing overhead associated with retransmissions (one extra idle cycle plus those cycles associated with the repetition of a basic read/write transfer) can be tolerated, we now investigate their energy efficiency compared to that of correction oriented strategies.

5. Error detection capability

In order to analyze the different proposed error control schemes, we fix a constraint on the residual word error probability, that is we impose that each scheme has the same probability of an undetected error at the decoder side. Should the wrong word not be captured, we assume that the system crashes.

The cost each scheme has to take on to be able to satisfy this common constraint is different, because they have different error detection capabilities. To verify this, we assume, as in [5], that every time a transfer occurs across a wire, it can make an error with a certain probability ϵ, where the value of ϵ depends on the supply voltage V_{dd} and the variance σ_N^2 of the noise voltage V_N. Hence, we derive the requirement on the maximum value of ϵ that can be tolerated by each scheme in order to ensure the common predefined level of reliability. In doing this, we assume the statistical independence of bus lines. If P is the residual word error probability, A_i is the number of Hamming code words with weight i, for a 32 bit bus we have:

ORIG:

$$P_{OR} = [1 - (1 - \epsilon_{OR})^{32}] \tag{1}$$

SEC:

$$P_S = 1 - \sum_{i=0}^{1} \left(\begin{array}{c} 38 \\ i \end{array} \right) \epsilon_S^i (1 - \epsilon_S)^{38-i} \tag{2}$$

SECDED:

$$P_{SD} = \sum_{i=1}^{19} \left(\begin{array}{c} 39 \\ 2i+1 \end{array} \right) \epsilon_{SD}^{2i+1} (1 - \epsilon_{SD})^{39-2i-1} \tag{3}$$

ED:

$$P_{ED} = \sum_{i=1}^{38} A_i \epsilon^i (1 - \epsilon)^{38-i} \tag{4}$$

PAR:

$$P_{PR} = \sum_{i=1}^{16} \left(\begin{array}{c} 33 \\ 2i \end{array} \right) \epsilon^{2i} (1 - \epsilon)^{33-2i} \tag{5}$$

By inverting the above relations, for each scheme we get the line flipping probability ϵ as a function of P, as shown in Fig. 4. Of course, schemes with higher detection capabilities have also less constraints on ϵ, although they take different recovery actions. Note that the curves relative to SEC and PAR are almost identical. Even if PAR detects all errors on an odd number of bits, against the single error detection of SEC, the probability to have multiple errors is very small, and this explains why the two curves are almost overlapped.

6. Power supply requirements

The parameter ϵ depends on the knowledge of different noise sources and their dependance upon the supply voltage, and is therefore difficult to estimate. So, for purpose of statistical analysis, we model the sum of several uncorrelated noise sources affecting a bus line as a single gaussian noise source, and we make use of the model developed in [5]. We therefore assume that a signaling waveform has a certain noise voltage V_N onto it, distributed according to a normal distribution with variance σ_N^2. The error probability of each single line can be written as

$$\epsilon = Q(\frac{V_{dd}}{2\sigma_N}) \tag{6}$$

where $Q(x)$ is the gaussian pulse

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy \tag{7}$$

This model accounts for the decrease of noise margins (and hence for an increase of the line flipping probability ϵ) caused by a decrease of the power supply voltage.

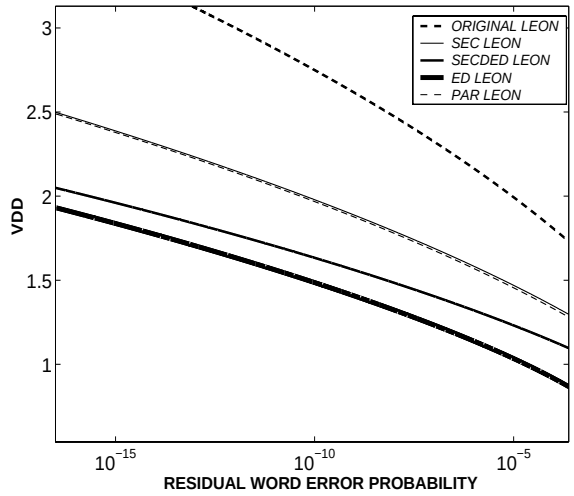

Figure 5. Requirements on VDD against those on line flipping probability

Given the requirements on line flipping probability of Fig. 4, we derive through the model the requirements on the power supply voltage for each error control scheme. The results are shown in Fig. 5, where we indicate the supply voltage V_{dd} and hence the swing on the bus lines that must be used to guarantee a predefined level of reliability. In the figure, the leftmost x-axis value of 10^{-15} corresponds to a time between failure (TBF) of 12 years, while for the rightmost residual word error probability the TBF is 4 ms (considering an AMBA bus clocked at 50 MHz).

Of course the original Leon processor, without noise tolerance, has the tightest requirements on V_{dd}, which must be large enough to prevent the violation of the reliability constraint. The other schemes can rely on their detecting capabilities and on recovery actions, thus allowing V_{dd}, and hence the noise margins, to be smaller.

Note that retransmission based techniques require lower V_{dd} than the correction based ones. This is mainly due to the fact that, for a linear code, the probability of a decoding error is much higher than the probability of an undetected error [8]. For instance, when a Hamming code is used for SEC, the decoder assumes that whenever an error occurs, it is single and not multiple, and can therefore proceed to correct it. Yet, it could have been also a multiple error, and in this case the corrected word is wrong. If no restrictive assumptions are made on the nature of a detected error, correction cannot be carried out but the detection capability improves a lot (see for example ED), and the supply voltage can be lower, as narrower noise margins can be tolerated. In this case, the system recovers from errors by means of retransmissions.

7. Power efficiency metric

The metric used to compute the power efficiency of the error control schemes under test is the average energy per useful bit, defined as

$$\bar{E}_{ub} = \frac{\sum_{i=0}^{m} p_i \bar{E}_i}{\sum_{i=0}^{m} p_i} \tag{8}$$

which is the weighted average of the average energy per useful bit $\bar{E}_i$ in those cases in which the system does not fail. p_i are the probabilities of occurrence of those cases. For example, SEC works well both in the error free case (it happens with probability p_0 and the average bus- and codec-related energy consumption per useful transferred bit is $\bar{E}_0$) and in the singol error case (probability p_1 and energy $\bar{E}_1$). For SECDED, ED and PAR we consider all detectable patterns, too. As to ORIG, only the error free case is taken into account.

In the above defined metric, the denominator represents the probability that the system works properly, and has the same value for all of the schemes, as we have fixed from the beginning a predefined level of reliability that the system must ensure.

The average energies $\bar{E}_i$ can be obtained, for each scheme, as follows:

$$\bar{E}_i = Eb_i + Ee_i + Ed_i \tag{9}$$

where Eb_i is the average energy per useful bit spent for bus transitions, while Ee_i and Ed_i express the energy consumption of encoder and decoder respectively.

All values of the parameters needed to compute the power efficiency metric have been derived from cycle-accurate VHDL simulations of the Leon processor with the error control schemes implemented in it. In particular, running the Dhrystone Benchmark on top of the processor, clocked at 50 MHz, we have stimulated 77410 cache misses and data transfers on the bus.

7.1. Bus transitions

From the simulations, we counted the number of bus transitions, thus computing the related average energy per useful bit as follows:

$$Eb_i = \frac{0.5 C_L V_{dd}^2 N_{tr}}{n N_{tf}} \tag{10}$$

where N_{tr} is the total number of transitions on the bus, N_{tf} is the total number of tranferred words (77410), n the number of useful bus lines (32), and C_L the bus line load capacitance. In this way, we consider the redundancy due to encoding-related extra bus lines and their transitions as an overhead that weighs on the standard useful bus lines.

System			Transfers	Transitions	Extra	Area		Power (uW)		Delay (ns)	
Scheme	Error	Err. recovery		Eb_i	Bus Lines	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.
ORIG	Free	-	77,410	768,949	-	0	0	0	0	-	-
SEC	Free	-	77,410	1,001,540	6	5,022	11,034	153	233	1.61	4.56
	Single	Correction	77,410	1,103,658				153	279		
SECDED	Free	-	77,410	1,041,672	7	6,588	14,238	205	308	2.31	4.85
	Single	Correction	77,410	1,143,196				205	360		
	Double	Retransmission	76,264	2,830,032*				254	363		
ED	Free	-	77,410	1,001,540	6	5,022	5,049	153	146	1.61	1.73
	Single	Retransmission	76,264	2,617,903*				190	144		
	Double	Retransmission	76,264	2,733,466*				190	148		
	Triple	Retransmission	76,264	2,848,762*				190	152		
PAR	Free	-	77,410	806,332	1	2,538	2,592	61	63	1.40	1.45
	Single	Retransmission	76,264	2,150,837*				77	62		
	Triple	Retransmission	76,264	2,386,697*				77	66		

*Including 98.397 address bus extra-transitions

Table 1. Average number of bus transitions and codec characteristics for each error control scheme

The values for Eb_i are reported in TABLE 1. While in ORIG we only considered the error free case, for the other schemes we had to consider Eb_i also for single-error cases (SEC, SECDED, ED, PAR), double-error cases (SECDED, ED) and finally for other multiple-error cases (PAR, ED) whose probability is meaningful. Errors on the bus have been reproduced by randomly flipping a desidered number of lines in the VHDL simulation. For those techniques that retransmit wrong data words, we assume the retransmitted word not to be in error. As TABLE 1 indicates, retransmission based strategies exhibit a larger number of transitions. In their count, we also considered the extra-transitions occurring on the address bus because of the early broadcast of the address of the next transfer, that will however be interrupted to allow the retransmission of the previously incorrectly received word.

The total number of transfers N_{tf} is slightly smaller for retransmission based systems because, during many burst transfers, branch or call instructions are read in. In this case, only the next transfer of the burst is completed. In the error free case, the burst completes on a new data word cache storage, while if the last word was affected by errors, the last transfer will be a retransmitted word. For this very particular kind of burst, the error free case exhibits one more transferred data word.

7.2. Codec overhead

Encoders and decoders for the different error control schemes are based on exor trees of different complexity and have been synthesized with Synopsys Design Compiler, using the CORELIB library from ST Microelectronics (V_{dd} = 2.5 V, L = 0.25 μm). A gate level simulation, combined with the use of Power Compiler, provided the power consumption of these gates under different working conditions.

Results are reported in TABLE 1, where it is evident that power consumption values reflect codecs' complexity. The simplest encoder is of course that of PAR, followed by SEC and ED. SECDED has the most complex one, mainly because it has to generate one parity bit more than SEC and ED.

Among the decoders, the extremely low complexity of PAR accounts for the lowest consumed power. SECDED exhibits a higher power respect to SEC, because of the circuitry for double error detection and for the retransmission triggering, while ED consumes much less thanks to the lack of any correction circuitry.

Looking at codec delays in TABLE 1, we notice that correction oriented strategies have the decoders with the longest critical path. This may result in additional wait states each time the memory is accessed, hence affecting the processor performance. On the contrary, detection oriented decoders incur less delay, and hence less timing overhead on each memory access, but their impact over performance is bigger in presence of erroneous transfers, when a retransmission is scheduled.

8. Energy per useful bit

Based on the simulation results and for all of the noise tolerance schemes, we express the average energy for useful bit as a function of the initial constraint on decoder residual word error probability. As we know from Fig. 5, each scheme is likely to meet the reliability constraint using different supply voltages V_{dd}. Therefore, the values of energy consumption derived in the previous section for a fixed V_{dd} of 2.5 V have been scaled to the proper supply voltages in order to evaluate power efficiency.

The results were derived under two different conditions:
- constraint on reliability only;
- constraint on reliability and power supply voltage;

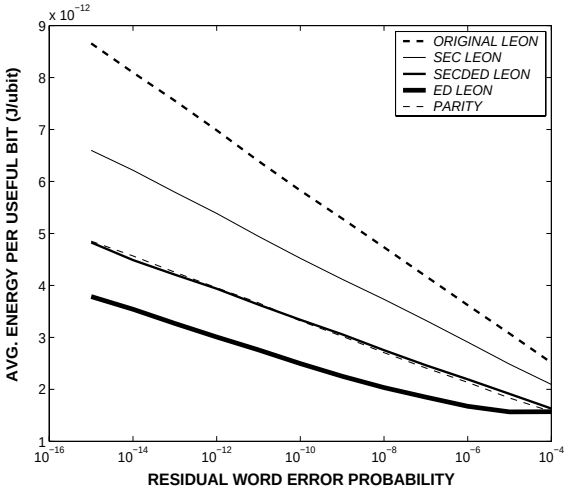

Figure 6. Energy efficiency for long wires

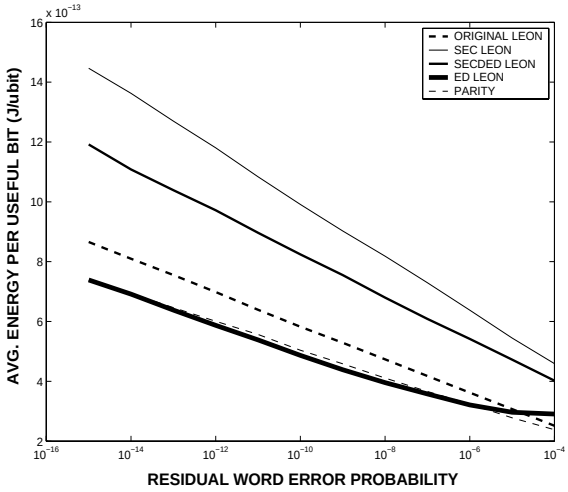

Figure 7. Energy efficiency for short wires

8.1. Constraint on Reliability

In this case we assume that power supply voltage is not a constraint. Hence, given the probability of system failure we are available to tolerate, we indicate the average energy per useful bit consumed by each scheme in order to meet that requirement. This metric takes into account both the detection capability of each code implementation and the recovery action adopted.

The results are reported in Figs. 6 and 7. The first plot refers to a bit line load capacitance C_L of 5 pF (a wire of about 1 cm in a 0.25 μm technology), while the second refers to a load of 0.5 pF (about 1 mm wire length).

In Fig. 6, the load capacitance per line being relatively high, the energy cost associated with bus transitions is dominant respect to encoders and decoders power overhead. The relative ratio of bus-related energy per bit over the codec-related one goes from 13 to 25, depending on the location on the plot.

If the reliability constraint is very tight (leftmost part of Fig. 6), retransmission based strategies (ED,SECDED) perform better than those using correction. Even though ED involves a much larger number of transitions than the original Leon processor ORIG, it can work at a supply voltage which is about 48% less (from Fig. 5), and considering the quadratic weight of the supply voltage in the computation of bus-related power supply, we can understand the large gap between the two techniques pointed out by Fig. 6. The other schemes lie in-between, confirming that techniques that can work at lower supply voltages because of their error detection capabilities, are also more effective from an energy viewpoint.

As we relax the reliability constraint (rightmost part of Fig. 6), the retransmission policy becomes less and less effective, because the other schemes can tolerate to work at lower supply voltages, and therefore their minor number of bus transitions starts making more and more the difference. ED can enjoy lower V_{dd} as well, but the slope of its curve in Fig. 5 is smaller than the competing schemes.

Note that PAR, in spite of its extremely low hardware complexity and overhead, does not perform better than ED, mainly because it is unable to detect double-errors.

Fig. 7 shows the same curves plotted for a C_L of 0.5 pF (local wires). Here bus transitions are less important while the impact of the codecs is no more negligible. ED still performs better, but also PAR reaches the same level of power performance. SEC and SECDED are the most power hungry approaches, because of their correction circuitry. ORIG, that can ensure reliability only by using proper V_{dd}, has no overhead due to coders and decoders, and therefore consumes less than SEC and SECDED. Yet, it still does not manage to make up for the detection capability of ED and PAR, only slightly counterbalanced by the overhead of their relatively simple codecs.

Moving from Fig. 7 to Fig. 6, we note a reduction of the gap between the energy efficiency of error correction and error detection techniques. The power cost for communication on the bus is expected to increase because of the larger load capacitances. At the same time, power associated with logic blocks will decrease, thus causing correction based techniques to progressively bridge the gap.

8.2. Constraint on Reliability and V_{dd}

Now we also assume that V_{dd} is fixed and cannot be used to select the most energy efficient scheme, as results from Figs. 6 and 7. The schemes that exhibit the lowest average energy per useful bit for this double-constrained problem are showed in Fig. 8, where well-defined regions appear.

For high V_{dd} (region A), all of the schemes meet the re-

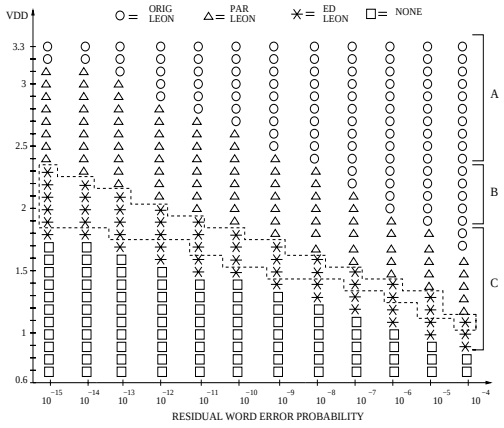

Figure 8. Choice of the most energy-efficient scheme under reliability and power supply constraints

liability requirements, even if with different margins, but ORIG is the simplest scheme with the smallest area and power overhead. When ORIG does not keep up with the constraint any more, PAR becomes the most effective solution because of its semplicity. If the system cannot tolerate the retransmission timing penalty induced by PAR, we can tradeoff power with speed by choosing SEC, the less consuming correction-based approach in this case.

Moving to region B, we observe that when PAR cannot satisfy the x-axis constraint any more, the less energy-per-bit consuming solution becomes ED. Yet, in the highlighted part of the plot, SECDED could be preferred, as it consumes more power but has less impact over execution speed.

Finally, in region C, as PAR does not meet the constraint any more, the only viable solution is ED that, outside the indicated area, cannot be traded off with SECDED any more. When even ED becomes ineffective, then other kinds of error detecting/correcting codes must be used.

9. Conclusion

This paper investigates how to deal with on-chip noise sources affecting data bus lines in an energy efficient way. We show that dealing with detected errors by requesting the retransmission of the incorrectly received data word is more effective than error correction, from an average energy-per-useful-bit viewpoint.

This result holds for the practical cases where the assumptions of statistical independence of bus lines and uncorrelation of noise sources affecting a single bit line are reasonable. If this is not the case, the residual error probability should be recomputed again in accordance with a different noise model.

Results were derived by implementing noise tolerance on the VHDL model of a SPARC V8 compliant processor, and by exploiting the slave retry response of AMBA bus standard to carry out retransmissions.

Acknowledgment

This work is supported in part by MARCO Gigascale Silicon Research Corp. and in part by NSF under grant CCR-9901190

References

[1] K. Aingaran, F. Klass, C. M. Kim, C. Amir, J. Mitra, E. You, J. Mohd, and S. K. Dong. “Coupling Noise Analysis for VLSI and ULSI Circuits”. *IEEE ISQED 2000*, pages 485–489, March 2000.

[2] L. Anghel and M. Nicolaidis. “Cost Reduction and Evaluation of Temporary Faults Detecting Technique”. *DATE 2000*, pages 591–598, March 2000.

[3] L. Anghel and M. Nicolaidis. “Evaluation of Temporary Fault Tolerant IC Architectures to the Rescue of Nanometer Technologies”. *MEDEA A-401 2nd Public Workshop*, 2000.

[4] J. A. Davis et al. “Interconnect Limits on Gigascale Integration (GSI) in the 21st Century”. *Proceedings of the IEEE*, 89(3):305–324, March 2001.

[5] R. Hegde and N. R. Shanbhag. “Toward Achieving Energy Efficiency in Presence of Deep Submicron Noise”. *IEEE Trans. on VLSI Systems*, 8(4):379–391, August 2000.

[6] R. Ho, K. W. Mai, and M. A. Horowitz. “The Future of Wires”. *Proc. of the IEEE*, 89(4):490–504, April 2001.

[7] A. Krstic, Y. M. Jiang, and K. T. Cheng. “Pattern Generation for Delay Testing and Dynamic Timing Analysis Considering Power-Supply Noise Effects”. *IEEE Transactions on CAD*, 20(3):416–425, March 2001.

[8] S. Lin and D. J. Costello. *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, Engl. Cliffs, 1983.

[9] D. Liu et al. “Power Consumption Estimation in CMOS VLSI Chips”. *IEEE Journal of Solid-State Circuits*, 29:663–670, June 1994.

[10] C. Metra and M. Favalli. “Bus Crosstalk Fault-Detection Capabilities of Error-Detecting Codes for On-Line Testing”. *IEEE Trans. on VLSI Systems*, 7(3):392–396, Sept. 1999.

[11] C. Metra and B. Ricco’. “Optimization of Error Detecting Codes for the Detection of Crosstalk Originated Errors”. *DATE 2001*, pages 290–296, March 2001.

[12] R. L. Pickholtz. *Digital Systems Testing and Testable Design*. Computer Science Press, New York, 1990.

[13] K. L. Shepard and V. Narayanan. “Noise in Deep Submicron Digital Design”. *IEEE/ACM ICCAD-96*, pages 524–531, November 1996.

[14] C. Svensson. “Optimum Voltage Swing on On-Chip and Off-Chip Interconnect”. *IEEE Journal of Solid-State Circuits*, 36(7):1108–1112, July 2001.

[15] J. F. Wakerly. *Digital Design: Principles and Practices*. Prentice-Hall, Upper Saddle River, Third Edition 2000.

[16] http://www.gaisler.com. Gaisler Research Website.

[17] H. Zhang, V. George, and J. M. Rabaey. “Low Swing On-Chip Signaling Techniques: Effectiveness and Robustness”. *IEEE Trans. on VLSI Systems*, 8(3):264–272, June 2000.

[18] M. Zorzi. “Some Results on Error Control for Burst-Error Channels under Delay Constraints”. *IEEE Trans. on Vehicular Technology*, 50(1):12–24, January 2001.