

Chapter 1

ERROR CONTROL SCHEMES FOR ON-CHIP INTERCONNECTION NETWORKS: RELIABILITY VERSUS ENERGY EFFICIENCY

Davide Bertozzi

Luca Benini

Dipartimento Elettronica Informatica Sistemistica

University of Bologna (Italy)

{dbertozzi,lbenini}@deis.unibo.it

Giovanni De Micheli

Computer System Laboratory

Stanford University (USA)

nanni@stanford.edu

Abstract Solutions for combined energy minimization and communication reliability control have to be developed for SoC networks. These two critical design goals can be traded-off by means of redundant bus encoding. In this chapter, the theoretical framework for energy and reliability analysis is introduced and several error control and recovery strategies are investigated in a realistic SoC setting. Furthermore, the chapter provides guidelines to select the most appropriate error control scheme for a given reliability and/or energy efficiency constraint.

Keywords: communication energy, reliability, bus encoding

1. Introduction

As the integration of a large number of IP blocks on the same silicon die is becoming technically feasible, the design paradigm for future Systems-on-Chip (SoC) is shifting from device-centric to interconnect-centric. Performance and energy consumption will be increasingly determined by the communication architecture. Under very high IP integra-

tion densities, on-chip realization of interconnection networks (*Networks-on-Chip, NoC*) is emerging as the most efficient solution for communication.

The design of these NoCs represents an adaptation of the wide-area network paradigm, well known to the communication community, to the deep sub-micron (DSM) ICs scenario. In this context, micronetworks of interconnects can take advantage of local proximity and of a lower degree of non-determinism, but have to meet new distinctive requirements such as design-time specialization and energy constraints [1].

Energy dissipation is a critical NoC design constraint, particularly in the context of battery-operated devices, and will be the target of this chapter. The SIA roadmap projects that power consumption can marginally scale up while moving from 100nm to 50nm technology. At the same time, projected clock frequency and number of devices on-chip are increased significantly. These trends translate directly into much tighter power budgets. Voltage scaling, as predicted by the roadmap, is helpful in reducing power. Nevertheless, voltage scaling alone will not suffice, and specific design choices for low-energy consumption will be required.

Computation and storage energy greatly benefits from device scaling (smaller gates, smaller memory cells), but the energy for global communication does not scale down. On the contrary, projections based on current delay optimization techniques for global wires show that global on-chip communication will require increasingly higher energy consumption. Hence, communication-related energy minimization will be a growing concern in future technologies, and will create novel challenges that have not yet been addressed by traditional high-performance network designers.

The most efficient way to achieve high-speed energy-efficient communication is the employment of low swing signaling [2]. Even though it requires the design of receivers with good adaptation to line impedance and high sensitivity (often achieved by means of simplified sense-amplifiers), power savings in the order of 10x have been estimated with reduced interconnect swings of a few hundreds of mV in a 0.18 μ m process [3].

As technology scales toward DSM, the energy efficiency concern cannot be tackled without considering the impact on signal integrity. These two competitive issues are brought up by the scaling scenario and their interaction can be briefly summarized as follows:

- Low swing signaling reduces signal-to-noise ratio, thus making interconnects inherently sensitive to on-chip noise sources such as cross-talk, power supply noise, electromagnetic interferences, soft errors, etc. This sensitivity is increased by the reduction of receiv-

ing gates voltage noise margins as an effect of the decreased supply voltages.

- Coupling capacitance between adjacent wires is becoming the dominant component of interconnect capacitance. This has an impact not only on signal integrity, but also on power consumption, as most power consumed by interconnects is associated with switching of coupling capacitances (*coupling power*).

As a consequence, solutions for combined energy minimization and communication reliability control have to be developed for NoCs. The energy-reliability trade-off can be efficiently tackled by means of redundant bus encoding. Given a predefined reliability constraint for on-chip communication, different coding schemes, with their own error recovery strategies, can be compared in their ability to meet the requirements with the minimum energy cost.

This chapter introduces the theoretical framework for energy and reliability analysis and compares several error control and recovery strategies in a realistic SoC setting. Furthermore, the chapter provides guidelines to select the most appropriate error control scheme for a given reliability and/or energy efficiency constraint.

2. Communication reliability

The distinguishing challenge for NoC design will be to provide adequate *quality of service (QoS)* with a limited energy budget under strong technology limitations. QoS requirements include, but are not limited to, communication reliability and performance.

With present technologies, most chips are designed under the assumption that electrical waveforms can always carry correct information on chip. As we move to consider DSM NoCs, communication is likely to become inherently unreliable because of the increased sensitivity of interconnects to on-chip noise sources. The most relevant ones are hereafter briefly described.

1 CROSSTALK

Sidewall coupling capacitance is becoming the dominant component of interconnect capacitance. The relative ratio between cross-coupling capacitance and total wire capacitance already amounts to 70%, but is expected to achieve 80% in a 50 nm process.

Some projections show that for semiglobal interconnects the wire critical length at which the peak crosstalk voltage is 10% of the

supply voltage decreases almost an order of magnitude by 2014 (50 nm process), which will drastically increase the number of interconnects with significant crosstalk. Local interconnects suffer from the same problem, as they are scaling more in the horizontal dimension rather than in the vertical one [4], thus increasing the sidewall coupling area between adjacent wires.

Crosstalk noise in real buses is usually tackled at the physical layer by means of shielding: a grounded wire is inserted between every signal wire on the bus. The effectiveness of this technique is counterbalanced by the doubling of the wiring area, that might be impractical when routing resources are scarce.

2 POWER SUPPLY NOISE

A large number of "simultaneous" switching events in a circuit within a short period of time can cause a current-resistance (IR) drop on the voltage references [6].

A further worsening of the reference voltages is due to the chip and package inductances, responsible for the Ldi/dt noise contribution, which becomes relevant as an effect of the higher frequencies allowed by the scaled transistor sizes.

SPICE simulations show that a 10-15% voltage drop during a cell transition period can increase cell propagation delay by 20-30% [7]. Interconnects propagation delay can be affected as well because of the temporarily reduced driving capability of wire capacitances.

Common solutions to relieve the power supply noise problem include topology optimization [9], wire sizing [10], on-chip voltage regulation [11] and decoupling capacitance deployment [8, 12].

3 ELECTROMAGNETIC INTERFERENCE (EMI)

The drastically increased number of simultaneously switching transistors per die, combined with faster edges due to increasing clock rates, make EMI a serious concern. External electric and magnetic fields can be coupled into circuit nodes and corrupt signal integrity.

Many design techniques have been developed to meet EMC demands for VLSI circuits, including RC low pass filtering, scalability and temperature compensation for pad drivers [14].

4 INTERSYMBOL INTERFERENCE (ISI)

The smaller cross sections of interconnects in DSM technologies result in a big increase in resistive parasitics. The consequent RC behaviour leads to signal dispersion: if a transmitted pulse across the wire spreads out into other time slots, it causes ISI (the result

of memory in the channel which causes consequent symbols to overlap). This represents a fundamental limitation on the bandwidth of DSM interconnects [5].

A code-based approach has been proposed as a workaround for ISI-related limitations, wherein streams are encoded in such a way that isolated bits (i.e. bits taking value b in time slot n while bits in time slots $n - 1$ and $n + 1$ take the complementary value \bar{b}) are not transmitted [15].

5 OTHER NOISE SOURCES

Logic values of chip internal nodes can be flipped as an effect of charge injection due to alpha particles (emitted by packaging materials) or to thermal neutrons from cosmic ray showers. This results in a *soft error*, that used to be relevant only for large DRAMs, but is now conjectured to be a potential hazard for large SoCs as well [13].

Other noise sources are generally handled by means of statistic models: namely, crosstalk between perpendicular wires at adjacent levels (*interlevel*), *thermal noise*, *shot noise* (caused by the quantization of current to individual charge carriers), and *1/f noise* (originated by random variations in components giving rise to a noise that has equal power per decade of frequency). For most cases of practical interest, these noise sources can be modelled as gaussian white noise.

Finally, synchronization problems are worth mentioning. As clock frequencies are increasing, the time difference between events decreases, and this may lead to synchronization failures. For instance, sampling changing data signals traveling across interconnects has to deal with the sampling uncertainty issue, combined with non-deterministic delays that may affect signal propagation. This might result in a sampling error. The same effect might be generated by *metastability*: simultaneous switching of circuit inputs might lead the outputs to a metastable state, wherein noise tends to make them converge in one direction or the other.

3. Traditional approach to fault tolerance

The self-checking capability of VLSI circuits can be defined as the ability to automatically verify whether there are faults in logic, without the need for externally applied test stimuli. Whenever this feature is available, on-line error detection is carried out, i.e. faults can be detected during the normal operation of a circuit.

Error detecting codes are widely deployed for the implementation of self-checking circuits (SCCs), mainly because of design cost considerations and because they allow error recovery to be carried out either in hardware or in software. Basically, a functional unit provides an information flow protected by an error detecting code, so that a checker can continuously verify the correctness of the flow and provide an error indication as soon as it occurs. Error correcting codes would on the contrary incur performance penalties related to additional correction circuitry.

Two frequently used codes in SCCs are *parity check code* and *two-rail code*. The former one adds only one parity bit to the information bits and detects all error patterns of an odd number of bits, but cannot detect double errors that can be relevant in a crosstalk-dominated scenario.

The two-rail code represents a signal as a pair of two complementary variables $(x_i, x_{i'})$, thus doubling the number of bus lines. This overhead may not always be acceptable in spite of the high error detection efficiency provided by the code [18].

It has been observed that many faults in VLSI circuits cause unidirectional errors (i.e. 0-1 or 1-0 errors, provided the two kinds of errors do not occur simultaneously). Therefore, coding schemes targeting this kind of errors are well-known to the testing community, such as *m-out-of-n code* and *Berger code*.

In particular, the Berger code is the optimal separable all-unidirectional error detecting code: no other separable code can detect all unidirectional errors with a fewer number of check bits [16]. The check bits are the binary representation of the number of 0s counted in the information bits.

As technology scales toward DSM, traditional schemes used in SCCs may lose their detection effectiveness when applied to on-chip buses, because of the new noise sources that come into play. Unidirectional errors cannot efficiently describe the effects of these noise sources, and the detection capability of multiple bidirectional errors instead of unidirectional errors will be the distinctive feature of error control schemes for DSM NoCs.

For instance, crosstalk may inherently give rise to bidirectional errors, when two coupled lines switch in the opposite direction and both transitions are delayed inducing sampling errors.

Many solutions have been proposed to overcome the detection capability limitation of traditional error control schemes with respect to multiple bidirectional errors:

- Acting on the layout in both a code-independent way (i.e. spacing rules, shielding, line crossing, etc.) or code-driven way (e.g. keeping the two complementary bits as far apart as possible in a

two rail code). Alternatively, layout information can be exploited to come up with weight-based codes, i.e. extensions of Berger or m-out-of-n codes that are able to deal more efficiently with bidirectional errors [19, 20].

- Acting at the electrical level (e.g. the probability of single errors can be increased with respect to that of bidirectional ones by unbalancing bus lines drivers).
- Using suitable detectors capable to deal with the effects of specific errors (e.g. crosstalk induced errors), but they might not be available in some design styles.

The major drawback of the above mentioned approaches is the need to have layout knowledge or to act at the electrical level. A more general approach could be desirable, wherein the proper course of action against multiple bidirectional errors can be taken early in the design stage, independent of the technology and the final layout, the knowledge of which is not generally available in advance.

Redundant bus encoding remains the most efficient approach for this purpose. Yet, new codes other than traditional ones must be used, targeting a wider class of errors than unidirectional ones. Linear codes could be a viable solution, in that they target error multiplicity rather than error direction. Moreover, their encoders can exhibit very lightweight implementations and can be provided with optional correcting capability. In the following section some details are given about a famous class of linear codes (Hamming code), whose flexibility and optimality in the number of parities make it suitable for micro-network applications, and about a class of linear codes with interesting characteristics for the same class of applications (cyclic codes).

4. Linear codes

In block coding, the binary information sequence is segmented into message blocks of fixed length; each message block, denoted by u , consists of k information digits. There are a total of 2^k distinct messages. The encoder, according to certain rules, transforms each input message u into a binary n -tuple v with $n > k$. This binary n -tuple v is referred to as the code word of the message u . Therefore, corresponding to the 2^k possible messages, there are 2^k code words. This set of 2^k code words is called a block code. For a block code to be useful, the 2^k code words must be distinct. A binary block code is *linear* if and only if the modulo-2 sum of two code words is also a code word [21]. The block code given in Table ?? is a (7,4) linear code.

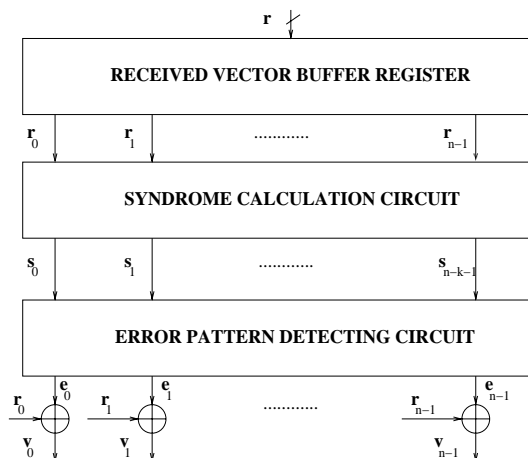


Figure 1.1. Decoder for a Hamming code.

4.1 Hamming codes

Hamming codes have been the first class of linear codes devised for error correction and have been widely employed for error control in digital communication and data storage systems.

For any positive integer $m \geq 3$, there exists a Hamming code [21] with the following parameters:

Code length: $n = 2^m - 1$

Number of information symbols: $k = 2^m - m - 1$

Number of parity-check symbols: $n - k = m$

Error-correcting capability: $t = 1$.

The decoder for a Hamming code is reported in Fig. 1.1. The whole circuit can be implemented as an EXORs tree. Note that the final correction stage is optional, and this makes Hamming code very flexible from an implementation viewpoint. According to the codec design, several versions of the Hamming code can be obtained: single error correcting code (SEC), single error correcting and double error detecting code (SECDED) and error detecting Hamming code (ED). This latter uses Hamming code for detection purposes only, and can therefore exploit its full detection capability, which includes not only all single and double errors, but also a large quantity of multiple errors (only those error patterns that are identical to the nonzero code words are undetectable).

Hamming codes are promising for application to on-chip micrometworks because of their implementation flexibility, low codec complexity

and multiple bidirectional error detecting capability. Note however that when correction is carried out, the detection capability of the code is reduced, because restrictive assumptions have to be made on the nature of the error. This explains why for a linear code the probability of a decoding error is much higher than the probability of an undetected error [21].

4.2 Cyclic codes

Cyclic codes are a class of linear codes with the property that any code word shifted cyclically (an end-around carry) will also result in a code word. For example, if $c_{n-1}, c_{n-2}, \dots, c_1, c_0$ is a code word, then $c_{n-3}, \dots, c_0, c_{n-1}, c_{n-2}$ is also a code word.

Cyclic redundancy check (CRC) codes are the most widely used cyclic codes (e.g. in the computer networks domain to guard the integrity of messages), and the new DSM scenario could raise the interest for their on-chip implementation, as will be hereafter briefly described.

An (n,k) CRC code is formed using a generator polynomial

$$G(x) = g_m x^m + g_{m-1} x^{m-1} + \dots + g_0 x^0$$

of degree $m = n - k$ and having $g_m = g_0 = 1$. If a data message of arbitrary length is expressed as a polynomial $M(x)$, the division of $x^m M(x)$ by $G(x)$ generates a remainder $R(x)$ of degree no more than $(m - 1)$. The m coefficients of the remainder are the check bits which are appended to the message. The resulting codeword $T(x)$ is then represented as

$$T(x) = R(x) + x^m M(x) \quad (1.1)$$

The highest degree bits are transmitted first (information bits), check bits last.

A cyclic code with m check bits has two relevant properties [22]:

- All error bursts of length less than or equal to m will be detected. The length of a burst is the span from first to last error, inclusive.
- All patterns of 1,2 or 3, or indeed, any odd number of errors will be detected.

Cyclic codes such as CRC codes can play a major role in detecting crosstalk related faults in DSM buses or micronetworks, thanks to the notion of burst error applied to the space domain. Capacitive coupling is more significant among contiguous lines, therefore crosstalk-induced errors are more likely to occur in neighboring wires. On the contrary, bursts of errors in the time domain regard errors affecting a certain

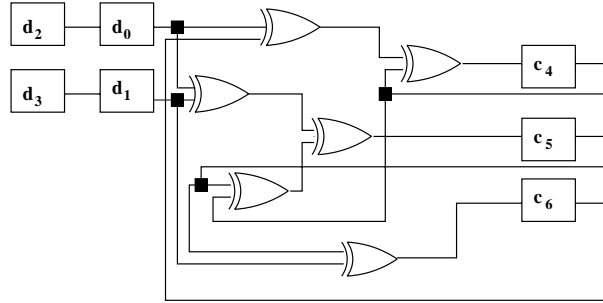


Figure 1.2. Serial-parallel PBCG implementation executing in 2 clock cycles. The transition matrix is raised to the power 2.

number of contiguous bits transmitted on the same line, but in DSM ICs this scenario is not relevant, except for ISI-related errors.

Hamming code targets error patterns rather than error bursts, and this may represent a useless redundancy: in fact, many error patterns detected by a Hamming code (e.g. multiple errors that are sparse all over the bus) may occur with an almost null probability in a real on-chip scenario, unlike the kind of errors detected by a cyclic code.

Cyclic redundancy checking can be easily carried out by means of a Linear Feedback Shift Register (LFSR), consisting of $n - k$ flip-flops: their state values are called "syndrome vector". This serial implementation is not clearly suitable for high-performance parallel communication, since it takes k clock cycles to execute.

The most efficient way to get parallel CRC implementations makes use of series-parallel sequence generation theory adapted to cyclic codes, and enables the maximum possible speed for a bounded requirement on circuit complexity [24]. The LFSR operation can be described by means of a recurrence equation that relates the syndrome vectors at times i and $i + j$ through a transition matrix T :

$$s_{i+j} = s_i T^j, j = 1 \dots k \quad (1.2)$$

where k is an integer multiple of j . By raising T to the power j we get a corresponding encoding/decoding circuit that executes in k/j clock cycles instead of k (see Fig. 1.2). This approach, called "Parallel Bit Code Generator" (PBCG), can be pushed to the limit by raising T to the power k , obtaining a combinational circuit that executes in one clock cycle at the cost of more circuit complexity. This technique provides a general strategy for identifying architectures with the desired speed-complexity trade-off.

System		Extra	Area		Power (uW)		Delay (ns)	
Scheme	Error	Bus Lines	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.
ORIG	Free	-	-	-	-	-	-	-
SEC	Free	6	5,022	11,034	153	233	1.61	4.56
	Single				153	279		
SECDED	Free	7	6,588	14,238	205	308	2.31	4.85
	Single				205	360		
	Double				254	363		
ED	Free	6	5,022	5,049	153	146	1.61	1.73
	Single				190	144		
	Double				190	148		
	Triple				190	152		
PAR	Free	1	2,538	2,592	61	63	1.40	1.45
	Single				77	62		
	Triple				77	66		
CRC-4	Free	4	2,376	2,637	54	62	0.65	1.02
	Single				68	61		
	Burst				68	65		
CRC-8	Free	8	2,160	2,700	47	58	0.39	0.88
	Single				59	59		
	Burst				59	62		

Table 1.1. Characteristics of synthesized codecs for different Hamming and CRC codes, and their average performance in those cases wherein communication completes successfully.

A comparison between synthesized codecs is shown in Table 1.1. Considering a 32 bit encoded bus, Hamming codes are compared with two CRC codes of generator polynomials $x^4 + 1$ and $x^8 + 1$. All of the error control schemes use retransmission as error recovery strategy, except for SEC and SECDED that correct single errors. A 0.25 μm synthesis library has been used, with a supply voltage of 2.5 V. Note that CRC codes exhibit the most lightweight implementations, comparable to that of a single parity check code (PAR).

5. Energy-reliability trade-off

The reliability issue that has been extensively discussed so far is tightly related to another fundamental constraint in NoC design: energy consumption. This is the major design constraint for SoCs installed on battery-operated devices such as handhelds, PDAs, etc..

With reference to on-chip buses, encoding strategies have been successfully exploited by the low-power community to save energy by reducing the switching activity on long wires (*low power encoding*). As technology scales toward DSM, coupling effects become more effective. This consideration has led to the development of *energy efficient* and

coupling driven bus encoding schemes, that consider coupling power in their power minimization framework. Yet another approach is viable, that consists of employing linear codes to meet predefined requirements on communication reliability, and of minimizing energy consumption by reducing interconnects voltage swing.

5.1 Low power encoding

An early idea to minimize bus transitions was to transfer an inverted word through the bus whenever it reduces the Hamming distance between two successive bus transitions. An additional bus line is used to indicate whether the word is inverted or not [25]. This technique, called *Bus-Invert Coding*, has been used as the reference point for a number of extensions and changes [26].

Low power bus encoding systems have also been split in a source encoding part and a channel encoding part [29]. The exploitation of source properties has been studied in [27] who proposed *Working-Zone Encoding (WZE)* to exploit locality of memory references. The underlying observation is that applications typically favour a few "working zones" of their address spaces. If both sender and receiver keep a table of base addresses of these working zones, an address can be expressed as an offset along with an index of the working zone based address. Later, in [28], the WZE was extended to multiplexed data buses.

A number of other techniques have been proposed, such as *transition pattern encoding* [30], *codebook-based encoding* [31], *probability-based mapping* [32] and *entropy-reducing code* [29].

5.2 Coupling-driven bus encoding for low power

A number of new bus encoding schemes (*coupling-driven signal encoding*) have been developed to alleviate coupling effects for DSM micronetworks or global on-chip interconnects. A simple solution is to minimize the occurrence probability of critical data patterns, wherein adjacent wires switch in the opposite direction.

An extended version of bus-invert coding can be used for this purpose: while the original technique flips the data signal when the number of switching bits is more than half of the number of signal bits, the coupling-driven bus-invert coding inverts the input vector when the coupling effect of the inverted signals is less than that of the original signal. The coupling effect is estimated by categorizing the transitions between adjacent wires and assigning a binary encoding to each of them (e.g. "00" for lines switching in the same direction, "11" for opposite switching directions, "01" for a single line switching, etc..). Then a majority

voter recognizes the number of "1"s in the encoded codeword, and if it is more than 50% of the total number of bits, the original data word is inverted due to the excess of critical transitions [35].

Another approach makes use of *dictionary encoding* techniques [33]. The key idea is to exploit a certain amount of correlation between adjacent bits of a word sent across the data bus when using real-world and freely available applications. Since there is not sufficient knowledge on data characteristics to build static dictionaries, adaptive dictionaries are used. An original data word is divided into three parts: non-compressed, index and upper part. The lower parts of the data words on data buses typically change quite frequently (i.e. lower correlation), so encoding should not affect the lower part. The encoder uses the index part of the current data to look-up the dictionary. Whenever there is a match (i.e. the upper part of the data word is in the dictionary), it then transmits the index part of the data word and the non-compressed part, so that the decoder is able to recover the upper part from its dictionary. Whenever a dictionary miss occurs, the encoder transmits the unchanged data word.

Using bus encoding to minimize coupling energy, which dominates the total bus energy, has the advantage of being a technology-independent approach that tackles the problem at the architectural level, but it must be observed that the increase in communication reliability comes only as a side-effect: it cannot be precisely quantified and noise sources other than cross-talk tend to be neglected. For DSM micronetworks, where the communication is expected to be inherently unreliable, tighter requirements on reliability will have to be met at a very limited energy cost, and the scenario could not necessarily be crosstalk-dominated.

5.3 Linear codes with low swing signaling

An alternative approach is to apply linear (cyclic) coding to on-chip micronetworks to meet predefined requirements on communication reliability, and to simultaneously minimize energy consumption by using low-swing signaling. Note that the reduction of the voltage swing across interconnects determines a decrease of signal-to-noise ratio (SNR), and hence an increased sensitivity to noise sources. Therefore, a trade-off exists between communication reliability and energy efficiency, as depicted in Fig. 1.3. The lower the probability of a codeword being transmitted in error, the higher the energy cost that has to be sustained by the communication architecture: very high detection capabilities have to be ensured by more complex codecs, and wire voltage swings have to be kept high to preserve high SNR values.

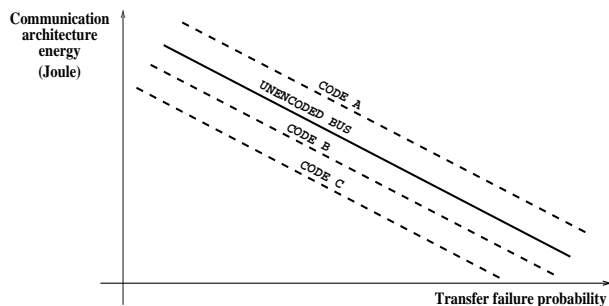


Figure 1.3. Reliability versus energy efficiency for redundant bus encoding schemes making use of low swing signaling.

The energy efficiency of a code is a measure of its ability to achieve a specified communication reliability level with minimum energy expense. This efficiency depends on a number of parameters, such as code detection capability, number of induced bus transitions, number of redundant bus lines, encoder and decoder implementation complexity. Therefore, each bus coding scheme is able to meet the reliability constraints with different energy costs, according to its intrinsic characteristics, and this allows to search for the most efficient code from an energy viewpoint (see Fig. 1.3).

An energy efficiency metric can be defined in order to make a comparison between different coding schemes [36]: the *average energy per useful bit*,

$$\bar{E}_{ub} = \frac{\sum_{i=0}^m p_i \bar{E}_i}{\sum_{i=0}^m p_i} \quad (1.3)$$

where p_i is the probability of having i errors at the same time affecting the transfer of a codeword. Not all values of i are considered in the metric, but only those ones corresponding to error patterns that are detectable by a certain error control code. E_i is the average energy consumed by the coding scheme implementation in the event described by p_i . All average energies are referred to a single useful bit: therefore the energy overhead associated with redundant parity lines is ascribed to the information lines, thus considering the impact of coding efficiency on energy efficiency.

As an example, Hamming SEC works properly both in the error free case (it happens with probability p_0 and the average bus- and code-related energy consumption per useful transferred bit is \bar{E}_0) and in the single error case (probability p_1 and energy \bar{E}_1).

In the above defined metric, the denominator can be thought of as the probability of correct operation of the system, and has the same value for all of the schemes that have to be compared, as it represents the common predefined communication reliability requirement.

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

$$\bar{E}_i = \bar{E}b_i + \bar{E}e_i + \bar{E}d_i \quad (1.4)$$

where $\bar{E}b_i$ is the average energy per useful bit spent for bus transitions, while $\bar{E}e_i$ and $\bar{E}d_i$ express the average energy consumption of encoder and decoder respectively. This figure of merit could be refined to account for bus coupling effects and their impact on energy consumption.

An important parameter needed to estimate energy efficiency of a code is the voltage swing used across interconnects. Its value is tightly related to the communication reliability, and a simple model has been proposed by Shanbhag [34]. The assumption is that every time a transfer occurs across a wire, it can make an error with a certain probability ϵ . The parameter ϵ depends on the knowledge of different noise sources and their dependence on the voltage swing, and is therefore difficult to estimate. So, for purpose of statistical analysis, the sum of several uncorrelated noise sources affecting a bus line is modelled as a single gaussian noise source, and the value of ϵ depends on the voltage swing V_{sw} and the variance σ_N^2 of the noise voltage V_N :

$$\epsilon = Q\left(\frac{V_{sw}}{2\sigma_N}\right) \quad (1.5)$$

where $Q(x)$ is the gaussian pulse

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy \quad (1.6)$$

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 voltage swing across a line, and allows a simplified investigation of the energy-reliability trade-off for the introduced linear codes.

As a case study, let us now consider the scenario depicted in Fig. 1.4. Bus encoding has been applied to the communication between a master (a SPARC V8 processor) and a slave (an on-chip memory) on a 32-bit AMBA bus (provided with retransmission capability), one of the most widely used shared bus architectures for present SoCs. Encoder and decoder are powered at standard voltage levels, while voltage level translators allow wires to work at a reduced swing. This setup points out the energy efficiency of the codes under test and provides useful

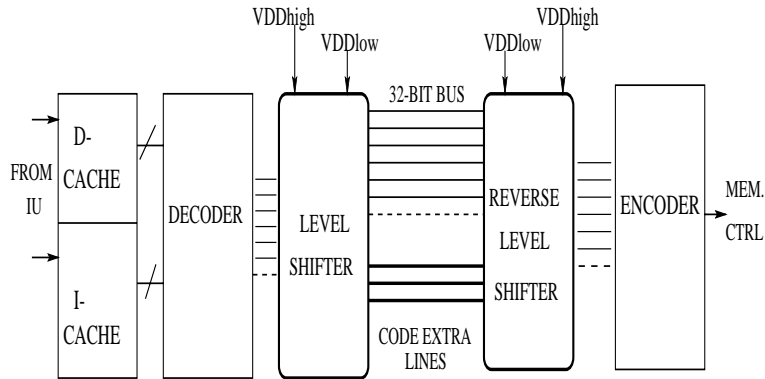


Figure 1.4. Realistic SoC setting where the energy-reliability trade-off is investigated.

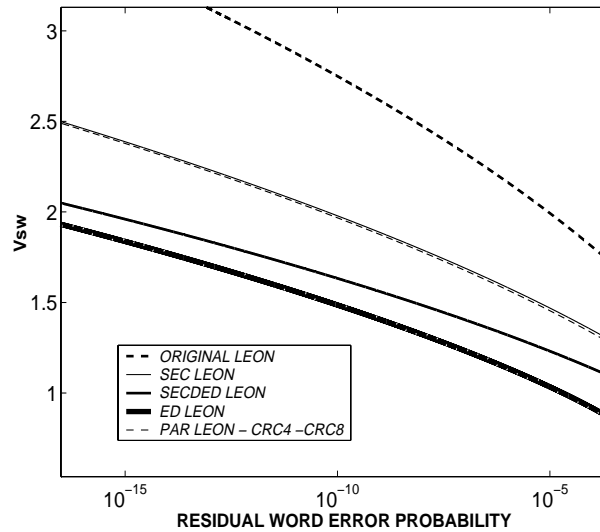


Figure 1.5. Minimum voltage swing needed by each coding scheme to meet a predefined communication reliability requirement.

indications for NoC designers because it can be thought of as a point-to-point connection in a NoC, e.g. between the network interface and a switch, or between two switches. For the purpose of our analysis, the only relevant difference respect to a multi-hop NoC scenario is that this latter makes use of data packetization, and this also affects the way retransmissions are carried out, as will be discussed in section 6.

Given the requirement on communication reliability on this bus, the minimum wire voltage swings V_{sw} that have to be used by error control codes to meet the common constraint can be derived by means of the Shanbhag model, and are reported in Fig. 1.5.

The unencoded bus has of course to use the highest swing, while the other encoding schemes can rely on their error detection capability, independently of the error recovery action. Note that retransmission based techniques, however, require lower swings than correction oriented ones, because they do not have to make restrictive assumptions on the nature of errors in order to be able to correct them. CRC codes have the same requirements as SEC and PAR, as they have the common characteristic of detecting single errors (having the highest occurrence probability with the used model) and not all double ones [37].

In general, the wider the detection capability of a coding scheme, the lower the voltage swing that can be used across interconnects: in fact, the lower SNR is counterbalanced by the ability of the decoder to detect a large number of error patterns and to take the proper course of recovery action. The next step is to evaluate whether such a swing reduction is beneficial in terms of energy dissipation: the energy overhead associated with error recovery must not make up for the low-swing related savings.

The impact of error recovery techniques can be assessed by computing the energy efficiency metric for all of the coding schemes. Results in Fig. 1.6 refer to a bit line load capacitance C_L of 5 pF (a wire of about 1 cm in a 0.25 μm technology). Such a load capacitance makes the energy cost associated with bus transitions dominant with respect to codec-related energy overhead. Retransmission based strategies (ED, PAR, CRC) perform better than SEC because they can work at lower voltage swings thanks to their higher detection capabilities, and this makes the difference independently of the increased number of bus transitions. Note that SECDED gives satisfactory results, in that it uses a mixed approach: correction is used for single errors, retransmission for double ones. Note that the leftmost part of the graph is the one of interest, because it corresponds to mean times between failures (MTBFs) in the order of years. In the rightmost part, MTBF is hundreds of milliseconds.

Fig. 1.7 shows the same curves plotted for a C_L of 0.5 pF (a few millimeter long wires). Here bus transitions play a minor role, while the contribution of codec complexity becomes relevant. This explains why the gap between SEC, SECDED and the other schemes becomes more relevant: correction circuitry at the decoder side makes the difference. Among retransmission-oriented schemes, PAR outperforms ED, and CRC4 and CRC8 become competitive.

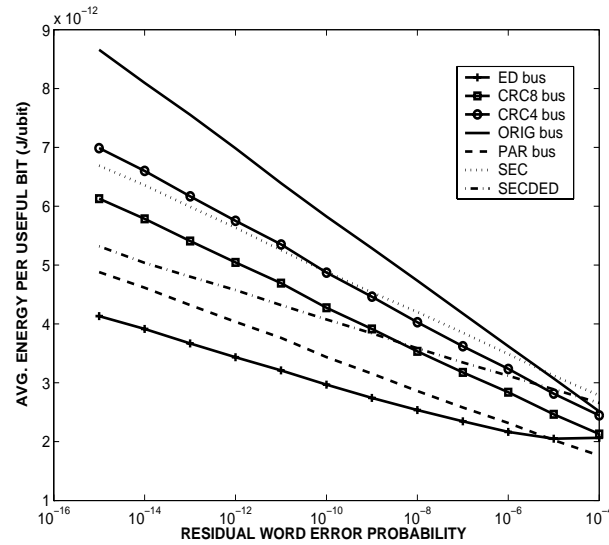


Figure 1.6. Energy efficiency of bus coding schemes for wire lengths in the the order of centimeters.

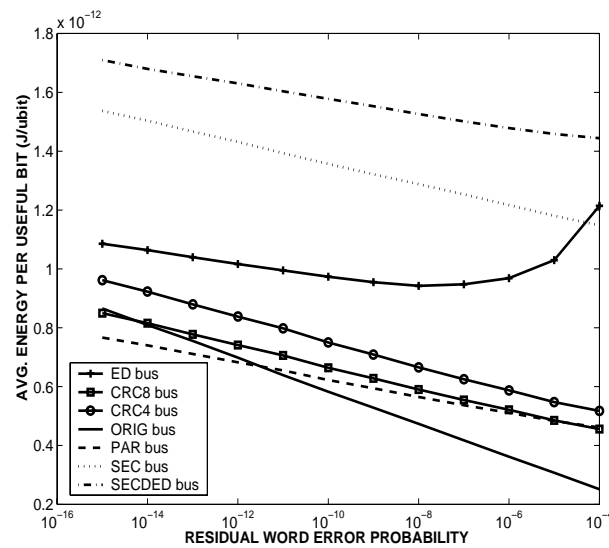


Figure 1.7. Energy efficiency of bus coding schemes for wire lengths in the order of millimeters.

The illustrated results point out that the detection capability of a code plays a major role in determining its energy efficiency, because it is directly related to the wire voltage swing. As regards the error recovery technique, error correction is beneficial in terms of recovery delay, but has two main drawbacks: it limits the detection capability of a code and it makes use of high-complexity decoders. On the contrary, when the higher recovery delay (associated with the retransmission time) of retransmission mechanisms can be tolerated, they provide high degrees of energy efficiency, thanks to the lower swings and simpler codecs (pure error detecting circuits) they can use while preserving communication reliability. Mixed approaches such as SECDED could be a trade-off solution.

6. Multi-hop NoCs

In multi-hop NoCs, the efficiency of retransmission also depends on network topology and on the relative distance (expressed as a number of hops) between sender and receiver. There are two possible scenarios:

- 1 The error recovery strategy can be *distributed* all over the network. Each communication switch is equipped with error detecting/correcting circuitry, so that error propagation can be immediately stopped. This is the only way to avoid routing errors: should the header get corrupted, its correct bit configuration can be immediately restored, preventing the packet from being forwarded across the wrong path to the wrong destination. Retransmission-oriented schemes also need buffering resources at each switch, so their advantage is more in terms of higher detection capability rather than circuit complexity, and the results derived throughout this chapter hold.
- 2 Alternatively, the approach to error recovery can be *concentrated*: only end-nodes are able to perform error detection/correction. In this case, retransmission may not be convenient at all, especially when source and destination are far apart from each other, and retransmitting corrupted packets would stimulate a large number of transitions, beyond giving rise to large delays. For this scenario, error correction is the most efficient solution, even though proper course of action has to be taken to handle incorrectly routed packets (retransmission time-outs at the source node, deadlock avoidance, etc.).

Another consideration regards the way retransmissions are carried out in a NoC. Traditional shared bus architectures can be modified to

perform retransmissions in a "stop and wait" fashion: the master drives the data bus and waits for the slave to carry out sampling on one of the following clock edges. If the slave detects corrupted data, a feedback has to be given to the master, scheduling a retransmission.

In packetized networks, data packets transmitted by the master can be seen as a continuous flow, so the retransmission mechanism must be either "go-back-N" or "selective repeat". In both cases each packet has to be acknowledged (ACK), and the difference lies in the receiver (switch or network interface) complexity. In a "go-back-N" scheme, the receiver sends a not ACK (NACK) to the sender relative to a certain incorrectly received packet. The sender reacts by retransmitting the corrupted packet as well as all other following packets in the data flow. This alleviates the receiver from the burden to store packets received out of order and to reconstruct the original sequence.

On the contrary, when this capability is available at the receiver side (at the cost of further complexity), retransmissions can be carried out by selectively requiring the corrupted packet without the need to retransmit also successive packets. The trade-off here is between switch (interface) complexity and number of bus transitions.

7. Conclusions

In this chapter, the energy-reliability trade-off has been investigated for SoC communication architectures, providing NoC designers with guidelines for the selection of energy efficient error control schemes. In particular it has been showed that:

- Error-control coding can significantly enhance communication reliability and temporarily reduce energy-per-bit dissipation. The energy overhead introduced by redundant bus lines is counterbalanced by the reduced voltage swings.
- The optimal encoding scheme is not unique but depends on bus loading conditions. For lines that are a few millimeters long, CRC codes are the most efficient solution because they target error bursts (better modeling the effects of crosstalk at a high abstraction level) with very lightweight implementations.
- With state of the art technology, retransmission turns out to be more efficient than correction from an energy viewpoint. However, as an effect of the IC scaling scenario, communication energy is likely to largely overcome computational energy, and for very DSM technologies error correction might bridge the gap.

References

- [1] Benini L., and De Micheli G. "Networks on chips: a new SoC paradigm" *Computer*, Vol.35, January 2002, pp.70-78.
- [2] Hui Z., George V., and Rabaey J.M. "Low-swing on-chip signaling techniques: effectiveness and robustness" *IEEE Transactions on VLSI Systems*, Vol.8, NO.3, June 2000, pp.264-272.
- [3] Svensson C. "Optimum voltage swing on on-chip and off-chip interconnect" *IEEE Journal of Solid-State Circuits*, Vol.36, NO.7, July 2001, pp.1108-1112.
- [4] Sylvester D., and Hu C. "Analytical modeling and characterization of deep-submicron interconnect" *Proceedings of the IEEE*, May 2001, pp.634-664.
- [5] Dally, W.J. and Poulton, J.W. (1998). *Digital Systems Engineering*. New York: Cambridge University Press.
- [6] Bakoglu, H.B. (1990) *Circuits, Interconnects and Packaging for VLSI*. MA: Addison-Wesley, 1990
- [7] Jiang Y.M., and Cheng K.T. "Analysis of performance impact caused by power supply noise in deep submicron devices" *Proceedings of Design Automation Conference*, June 1999, pp.760-765.
- [8] Chen H.H., Ling D.D. "Power supply noise analysis methodology for deep-submicron VLSI chip design" *Proceedings of Design Automation Conference*, June 1997, pp.638-643.
- [9] Erhard K.H., Johannes F.M., and Dachauer R. "Topology optimization techniques for power/ground networks in VLSI" *Proceedings of Design Automation Conference in Europe*, March 1992, pp.362-367.
- [10] Dutta R., and Sadowska M.M. "Automatic sizing of power/ground networks in VLSI" *Proceedings of Design Automation Conference*, June 1989
- [11] Ang M., Salem R, and Taylor A "An on-chip voltage regulator using switched decoupling capacitors" *Proceedings of int. Solid-State Circuits Conference Dig.Tech.Papers*, February 2000, pp.438-439
- [12] Zhao S., Roy K. and Koh C.K. "Decoupling capacitance allocation and its application to power-supply noise-aware floorplanning" *IEEE Transactions on CAD of Integrated Circuits and Systems*, Vol.21, NO.1, January 2002, pp.81-92.
- [13] Prince B. "Report on Cosmic Radiation Induced SER in SRAMs and DRAMs in Electronic Systems, May 2000.
- [14] Steinecke T. "Design-in EMC on CMOS large-scale integrated circuits" *2001 Int. Symposium on electromagnetic compatibility*, Vol.2, August 2001, pp.910-915.
- [15] Bogliolo A. "Encoding for high-performance energy-efficient signaling" *ISLPED*, August 2001, pp.170-175.
- [16] Pradhan, D.K. (1986). *Fault-Tolerant Computing: Theory and Techniques*. Englewood Cliffs, NJ: Prentice Hall.
- [17] Metra C., Favalli M., and Ricco' B. "On-line detection of bridging and delay faults in functional blocks of CMOS self-checking circuits" *IEEE Transactions on CAD*, Vol.5, NO.4, 1997, pp.770-776.
- [18] Favalli M., and Metra C. "Bus crosstalk fault-detection capabilities of error-detecting codes for on-line testing" *IEEE Transactions on VLSI Systems*, Vol.7, NO.3, September 1999, pp.392-396
- [19] Das D., and Touba N. "Weight-based codes and their application to concurrent error detection of multilevel circuits" *Proceedings of IEEE VLSI Test Symposium*, 1999, pp.370-376.
- [20] Favalli M., and Metra C. "Optimization of error detecting codes for the detection of crosstalk originated errors" *Proceedings of DATE 2001*, March 2001, pp.290-296.

- [21] Lin, S. and Costello, D.J. (1983) *Error control coding: fundamentals and applications*. Englewood Cliffs, NJ: Prentice Hall.
- [22] Mazo J.E., and Saltzberg B.R. "Error-burst detection with tandem CRC's" IEEE Transactions on Communications, Vol.39, NO.8, August 1991, pp.1175-1178.
- [23] Sobski A., and Albicki A. "Partitioned and parallel cyclic redundancy checking" Proceedings of the Midwest Symposium on Circuit and Systems, Vol.1, August 1993, pp.538-541.
- [24] Popplewell A., O'Reilly J.J, Williams S. "Architecture for fast encoding and error detection of cyclic codes" IEE Proceedings on Communications, Speech and Vision, Vol.139, NO.3, June 1992, pp.340-348.
- [25] Stan M.R., and Burleson W.P. "Bus-invert coding for low-power I/O" IEEE Transactions on VLSI Systems, Vol.3, NO.1, 1995, pp.49-58
- [26] Kretzschmar C., Siegmund R., and Mueller D. "Adaptive bus encoding technique for switching activity reduced data transfer over wide system buses" Int. Workshop - Power and Timing Modeling, Optimization and Simulation, September 2000.
- [27] Musoll E., Lang T., and Cortadella J. "Working-zone encoding for reducing the energy in microprocessor address buses" IEEE Transactions on VLSI Systems, Vol.6, NO.4, December 1998, pp.568-572.
- [28] Lang T., Musoll E., and Cortadella J. "Extension of the working-zone-encoding method to reduce the energy on the microprocessor data bus" Int. Conference on Computer Design, October 1998.
- [29] Ramprasad S., Shanbhag N.R., and Hajj I.N. "A coding framework for low-power address and data busses" IEEE Transactions on VLSI Systems, Vol.7, NO.2, June 1999, pp.212-221.
- [30] Sotiriadis P.P., and Chandrakasan A. "Bus energy minimization by transition pattern coding (TCP) in deep sub-micron technologies" IEEE/ACM Int. Conference on CAD, 2000, pp.322-327.
- [31] Komatsu S., Ikeda M., and Asada K. "Low power chip interface based on bus data encoding with adaptive codebook method" Great Lakes Symposium on VLSI, 1999, pp.368-371.
- [32] Benini L., Macii A., Macii E., Poncino M., and Scarsi R. "Architectures and synthesis algorithm for power efficient bus interfaces" IEEE Transactions on CAD ICs and Systems, Vol.19, NO.9, 2000, pp.969-980.
- [33] Lv T., Henkel J., Lekatsas H., and Wolf W. "An adaptive dictionary encoding scheme for SOC data buses" Proceedings of DATE 2002, March 2002, pp.1059-1064.
- [34] Hegde R., and Shanbhag N.R. "Toward achieving energy efficiency in presence of deep submicron noise" IEEE Transactions on VLSI Systems, Vol.8, NO.4, August 2000, pp.379-391.
- [35] Kim K.W., Baek K.H., Shanbhag N., Liu C.L., and Kang S.M. "Coupling-driven signal encoding scheme for low-power interface design" IEEE/ACM ICCAD, November 2000, pp.318-321.
- [36] Bertozzi D., Benini L., and De Micheli G. "Low power error resilient encoding for on-chip data buses" Proceedings of DATE 2002, March 2002, pp.102-109.
- [37] Bertozzi D., Benini L., and Ricco' B. "Energy-efficient and reliable low-swing signaling for on-chip buses based on redundant encoding" Proceedings of ISCAS 2002, May 2002, Vol.I, pp.93-96.