

Power Consumption of Fault Tolerant Codes: the Active Elements*

D. Rossi*, V.E.S. van Dijk†, R.P. Kleihorst†, A.K. Nieuwland†, C. Metra*

* DEIS, University of Bologna, Viale Risorgimento 2, 40136, Bologna, Italy

† Philips Research Laboratories, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands

Abstract

On-chip global interconnections in very deep submicron technology (VDSM) ICs are becoming more sensitive and prone to errors caused by power supply noise, crosstalk noise, delay variations and transient faults. Error correcting codes can be employed in order to provide signal transmission with the necessary data integrity. We compared Dual Rail encoding versus Hamming with respect to power consumption of the bus wires themselves (passive capacity model) [13]. In this paper we analyze the contribution of the active elements of both coding schemes. We first present a detailed analysis of the power consumption of an encoded bus, taking into account the bus wires (with mutual capacitances, drivers, repeaters and receivers), as well as the encoding/decoding circuitry. Then we compare the two considered coding technique with respect to the power consumption, and we show how different tradeoffs can be achieved. Our analysis is based on a realistic bus structure, implemented in a 0.13 μ m CMOS technology.

1 Introduction

As technology scales to very deep submicron (VDSM), noise affecting bus wires is becoming one of the major concerns for digital systems. In fact, the increased integration density, reduced node capacitances, power supply and noise margins, as well as the process parameter variations and the continuous increase in operating frequencies, make ICs more sensitive and prone to transient faults, crosstalk noise and delay variations [1, 8].

To guarantee an adequate signal integrity during on-chip communication, a fault tolerant bus can be adopted. In fact, for on-chip global interconnections, the reliability issue can be addressed by implementing techniques based on on-line testing and diagnosis, followed by proper fault recovery. For instance, an error due to noise affecting the bus

wires can be first concurrently revealed using a specific detector [6, 9, 2, 7], and then masked by data retransmission to achieve fault tolerance. Besides these approaches, either error detecting codes followed by proper recovery, or error correcting codes, which provide on-line correction and do not require retransmission, can be implemented [4, 13].

Another big issue of modern electronic systems is power consumption, both in logic and bus wires. In fact, as the operating frequency increases, the power dissipation increases as well. Consequently, a power-efficient design requires a low power dissipation in all parts of the design [3]. Several techniques have been proposed to reduce power consumption during bus transmission. They mainly rely on minimization of bus activity [11, 16], or on the reduction of the number of opposite transitions between adjacent wires [15, 14].

Trading off reliability and power consumption is a very challenging issue for the design and test community. This goal can be achieved by adopting a low swing signaling technique. For example, in [4], trade offs between energy efficiency and fault tolerance capability, provided by error detecting/correcting codes, have been investigated. These techniques rely on the observation that, once provided bus communication with an error correction ability (achieved by implementing either error correcting codes or error detecting codes followed by retransmission), sufficient signal integrity can still be provided although low swing signaling is used. These approaches, however, do not investigate the ability of an error detecting/correcting code to be “intrinsically” low power because of the characteristic of its codespace.

In [13] we presented an analysis of power consumption in Hamming codes, the most widely employed class of error correcting codes [12, 10, 5]. We showed that, by choosing among all possible Hamming codes with the same correction capability, no power optimization was possible. We then proposed a coding technique, that we called *Dual Rail*, which, combined with a proper bus layout, allows a significant power reduction with respect to Hamming codes.

In this paper we propose a detailed analysis on power consumption for both coding techniques (Hamming and

*This work is supported by the European Commission under IST-2001-38930 (FTEA)

Dual Rail) considering also the active elements, i.e., the buffers within the bus and the *codec* circuitry (i.e., the encoding and decoding circuitry). In fact, when the power consumption due to bus wires is reduced (for instance, using a suitable bus layout and/or the selecting a proper set of wire transitions), the power consumption due to the active elements of an encoded bus turns out to play an important role, giving a significant contribution to the total power dissipation.

This paper is organized as follows. In Section 2 we first introduce the considered bus model, then we show how to compute the power consumption of an encoded bus (including buffers), and how to optimize the inter-wire spacing in order to reduce the power consumption during bus activity. In Section 3, we introduce the codec schemes for both the considered codes. In Section 4, we present a detailed theoretical analysis of the power consumption due to both the codec circuits. Simulation results and their comparison with the theoretical calculations are reported in Section 5, while some conclusions are drawn in Section 6.

2 Bus Model and Power Consumption for the Considered Encoded Bus

In VDSM technology, the lumped wire model is not realistic to compute the power consumption on the bus during system's normal activity, because this model does not take into account any mutual effect between adjacent wires.

A good wire model, which does take mutual capacitances into account, is reported in [13]. This model is shown in Figure 1, where $C_B (F/m^2)$ is the contribution of the wire to the bottom parallel plate, $C_{EB} (F/m)$ is the contribution of the wire edge to the bottom (the fringing field) and $C_{EC} (F/m)$ is the wire to wire lateral component.

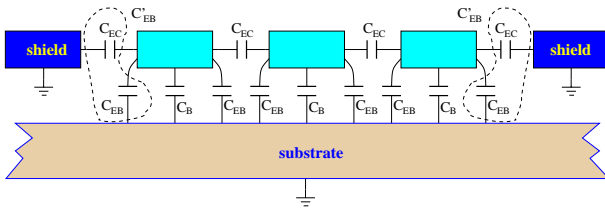


Figure 1. Parasitic capacitances of a 3 wire bus above substrate.

In [13] we presented a model to compute the power consumption of bus wires during bus activity. We showed that combining the *Dual Rail* approach with a proper bus layout, based on an intelligent spacing between adjacent wires, a consistent power reduction was possible with respect to the Hamming case.

Let us now define the following normalized parameters: $\lambda = C_{EC}/C_{BOTmin}$, $\gamma = C_{BOT}/C_{BOTmin}$ and

$\delta = C_{BOT}/C_{BOTmin}$, where C_{BOTmin} is the minimum value of the bottom capacitance, reached when the spacing between adjacent wires is minimum. For a *Metal2 – Substrate* shielded bus, implemented using a $0.13\mu m$ CMOS technology, considering the minimum spacing between wires, the following values are obtained: $\lambda = 7.3$, $\gamma = 8.3$ and $\delta = 1$. The power consumption during bus activity, considering only the bus wires, can be expressed as:

$$P_{Wire} = \{0.25[(n-2)\delta + 2\gamma] + 0.5(n-1)\lambda\} C_{BOTmin} V_{dd}^2 f, \quad (1)$$

where n , V_{dd} and f are the number of bus wires, voltage supply and frequency, respectively. The terms 0.25 and 0.5 represent the effective switching activity (*SA*) for the wire bottom and mutual capacitances, respectively [13].

A general correction scheme for a fault-tolerant bus is shown in Figure 2. The k information bits ($d_0 \dots d_{k-1}$) are given to an encoder (*E*) which calculates p check bits ($c_0 \dots c_{p-1}$). All the $n = k + p$ bits, which form a *code-word*, are transmitted on the bus. At the receiving end, the decoder (*D*) checks if an error occurred during transmission and generates k error bits ($e_0 \dots e_{k-1}$), which allows the corrector (*C*) to perform the right correction.

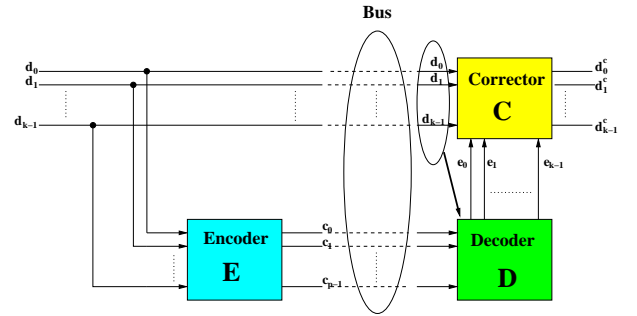


Figure 2. Schematic representation of a bus encoding/decoding scheme.

In order to reduce the power consumption within an encoded bus, we proposed a new coding technique, called *Dual Rail*. Our proposed code consists of adding $k + 1$ check bits ($c_0 \dots c_k$) to the original k information bits ($d_0 \dots d_{k-1}$). In particular, the check bits $c_0 \dots c_{k-1}$ are copies of the information bits, while c_k represents the data parity bit. Consequently, if the bits carrying the information and the respective copies are sent throughout adjacent wires, the respective mutual capacitances never need to be charged. Thus, *Dual Rail* allows us to reduce the number of mutual capacitances to be charged during bus activity. In fact, even though the number of bus wires increases with respect to the Hamming case, the smaller number of mutual capacitances which must be charged, together with the *intelligent spacing*, leads to a power consumption reduction

[13].

In order to reduce the propagation delay and to minimize the effect of crosstalk in the delay uncertainty, we have considered a bus model in which the wires are divided into shorter sections by the introduction of repeaters. In particular, we have considered a $10mm$ bus line divided into five parts, each $2mm$ long, by using 4 repeaters. All the repeaters have the same output impedance as the drivers, while the receiver at the end of a line is significantly weaker.

Typically, the power dissipated by a buffer consists of two components: input power dissipation and internal power dissipation. The input power dissipation is the power dissipated because of the input capacitance of the gate. Of course, the current is actually provided by the preceding gate. The internal power dissipation is due to the internal parasitic capacitances and the short circuit current. However, in a well-designed buffer, the short circuit dissipation is generally small. Instead, the internal power dissipation can be up to the 30% of the total power dissipation of the bus. Therefore, to compute the power consumption of the bus wires, we need to consider also the active elements (drivers, repeaters and receivers) of the bus.

The total power consumption of the active elements (drivers, receivers and repeaters) of one wire is:

$$P_{active} = 5(P_{inp,dr} + P_{int,dr}) + (P_{inp,rec} + P_{int,rec}), \quad (2)$$

where $P_{inp,dr}$ is the input power dissipation of the driver and 4 repeaters, $P_{int,dr}$ is the internal power dissipation of the driver/repeater. Similarly, we then have to consider the contribution of the receiver.

3 Proposed Hardware Scheme for Encoder and Decoder

Figure 3 shows the physical structures of the decoder and encoder circuits for the Hamming (a) and Dual Rail (b) approaches. The *Dual Rail* encoder has one data-wide parity tree connected to every input and build-up with XOR gates. In fact, only the parity bit c_4 has to be computed, since $c_0 \dots c_3$ are a copy of the corresponding data values. In the decoder, *Dual Rail* recomputes the parity of the data signals, compares it with the received parity and switches the multiplexers at the output to the copy signal if the computed and received parity differ from each other.

To verify the correctness of our proposed scheme, let us analyze all possible cases. If no error occurs or if a fault affects one of the check bits $c_0 \dots c_3$, we have $s_0 = 0$ and the data $d_0 \dots d_3$ are correctly transferred to the outputs of the multiplexers. In the case of an error involving a data bit or the check bit c_4 , it is $s_0 = 1$, and the multiplexers outputs are set to the values of the copy bits $c_0 \dots c_3$, which are correct. Hence, the proposed *ECC* circuit always behaves properly in the case of single bit errors.

The Hamming encoder (shown in Figure 3(a)) has more, but smaller, trees that are fixed to the selected inputs determined by the code construction matrix. In fact, all the parity bits (c_0, c_1, c_2) need to be computed from the data. In the decoder, the same parities are constructed from the incoming data signals and compared to the received parities, generating the error syndrome vector $(s_0, s_1, s_2)^T$. The syndrome is then decoded, in order to localize the possible faulty bit (datum or parity), by the 3 input AND gates, which provide the binary index of the dataline or parity line in error. After binary one-shot decoding (usually only the dataline error signals are decoded), the respective error correction XOR at the output is stimulated to flip the erroneous signal or to be transparent for correct signals.

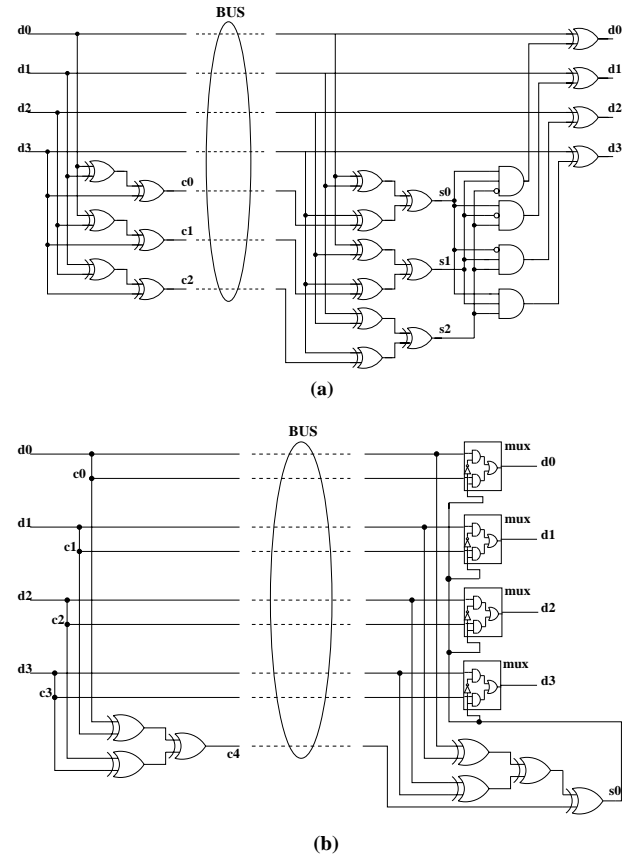


Figure 3. Encoder and decoder for (a) a 4 bit Hamming and (b) a Dual Rail system.

4 Energy consumption in Encoders and Decoders: Hamming versus Dual Rail

The error correcting encoders and decoders consume energy even in the error-free case by simply passing the data and checking the parities. Although the circuits for the Hamming and *Dual Rail* code are similar in structure, they

differ in number of gates, because of the different number of parity signals. In this section we analyze the structures and derive the energy consumption for the error free case in a semi-theoretical way annotating with numbers for the most recent IC technologies. We do not consider the case where the signal is influenced by errors, because we are still at the assumption that an error will seldomly occur, and therefore it will not have any significant influence on the global power consumption. The physical structures of these circuits are translated into basic gate equivalents with listed switching power consumption weighted by their activity for a maximum information rate on the input bus.

Looking only at the error-free case and ignoring decoder glitches, (induced by differences in the delay of the data, copy and parity signals), we can deduct that only the XOR gates in both parity trees and part of the switching multiplexers are active.

For a k bit Dual Rail (DR) encoder the active gates are $k - 1$ XORs. In the DR decoder, the active gates are $k - 1$ XORs for the parity tree, and the multiplexer box. The switching of the outputs of these gates due to normal data transitions will be part of the energy consumption of the decoder. With maximum information rates, these various gates in the complete codec have a switching activity of 50%.

The number of active gates for a Hamming setup is less straightforward to express. Hamming codes have to obey the following relationship between the number of parities p and the number of databits k :

$$p = \lceil \log_2 \{p + k + 1\} \rceil. \quad (3)$$

Typically Hamming codes are designed for minimal rate overhead, which equates to minimum parity as in $k_{max} = 2^p - p - 1$. However, we will derive the active gate count for any code with k data bits and p parity bits.

The total number of inputs to the parity trees, as employed for both the encoder and decoder for k data bits, follows a recursive function:

$$\#inputs(k) = \#inputs(k - 1) + \#cost(k), \quad (4)$$

where $\#cost(k)$ indicates the cost (number of additional inputs) by increasing the code from $k - 1$ to k data bits. This cost depends on where k is in relation to 1 and k_{max} . This set is arranged in several groups and the value of $\#cost(k)$ is equal to the groupnumber g that k is in:

$$\#cost(k) = g, \quad \text{when} \quad \sum_{i=2}^{g-1} \binom{p}{i} < k \leq \sum_{i=2}^g \binom{p}{i}. \quad (5)$$

For instance, for bits 1 to 6 of a 4 parity Hamming code, the cost of each of them is 2 additional inputs on the trees. For bits 7 to 10, the cost is 3 inputs, and for the final 11th

($k = k_{max}$) bit the cost is 4 inputs. The total number of XORs needed to build those trees equates to:

$$\#XORs = \#inputs(k) - p. \quad (6)$$

In addition to the parity decoder tree, the Hamming decoder consumes energy in the k error correction gates. All gates have an activity of 50% based on data sessions for maximum information throughput. The XOR gates of the decoder that compare the computed parities with the received parities are inactive for error-free transmissions, as well as the syndrome one-shot decoders. Because for typical applications power consumption is mainly dictated by the error-free case, we do not take the inactive gates in account.

We define the power consumption of switching a basic gate equivalent (a 2-input NAND) driving two similar gates, as *Equivalent Gate Power, EGP*. It is worth noticing that we defined this normalization factor in order to obtain results independent from the technology, and to more easily compare the achieved results. The 2-input AND and OR gates are seen as gate equivalents, while the XOR gates are seen as 2 basic gate equivalents, thus consuming twice as much, $2EGP$. Multiplexers consume $1.5EGP$.

Table 1. Power consumption of the encoders, decoders and correctors, expressed in EXOR power consumption.

Type	encoder	decoder	correction
Ham.	Eq. (6) · P_{XOR}	Eq. (6) · P_{XOR}	$k \cdot P_{XOR}$
DR	$(k - 1) \cdot P_{XOR}$	$(k - 1) \cdot P_{XOR}$	$k \cdot P_{MUX}$

The energy consumption of the Hamming corrector is the 25% higher than that of the dual rail corrector, because the Hamming corrector is build from XOR gates which consume more than the dual rail's multiplexer.

To compare the energy consumption of a k type DR and k type Hamming system, we first look at the energy consumption of the decoder and encoder part reported in Table 1.

The power consumption of both encoders and decoders, expressed in *EGP*, for some (small) values of k , is represented in Figure 4(a) and (b), respectively.

As we have already seen, for larger k the Hamming encoders and decoders have more active gates than the *Dual Rail* encoders and decoders, thus consuming significantly more energy than the comparable *Dual Rail* solutions.

5 Experimental Results and Comparison with Theoretical Results

Throughout this paper, we mainly focussed on a mathematical approach for power consumption in Hamming and

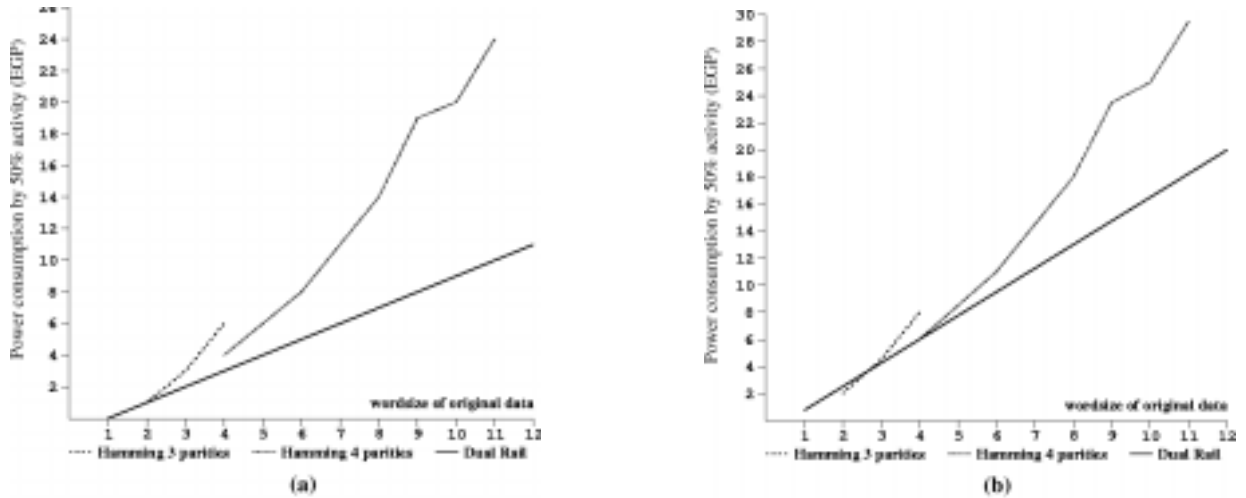


Figure 4. Power consumption for both encoders (a) and for both decoders (b).

Dual-Rail busses. In this section we present the experimental results and compare them with the mathematical figures. The experimental figures are obtained from simulations performed by means of HSpice. In our simulation file we included a distributed RC -model for the wires but, compared to [13], we also took all active elements into account (encoder, decoder, drivers, repeaters, receivers).

Our distributed RC -model incorporates both bottom and mutual capacitances. We built a 9-wire distributed RC -model for the Hamming bus and an 11-wire distributed RC -model for the Dual-Rail bus. The outer wires of both models are shield wires.

We simulated all possible transitions between all possible codewords in one simulation run, and listed the obtained power figures in Table 2, expressed in EGP . In our simulations we varied the total footprint (expressed in number of minimum wire pitches), and in particular we simulated each bus for $FP11$, $FP13$ and $FP16$, where in $FP13$ and $FP16$ we used intelligent wire spacing for the *Dual Rail* bus.

A footprint of 16 wire pitches was expected to be the optimum for the proposed power reduction technique. With respect to [13], the introduction of repeaters in the bus enables us to do simulations using a clock speed of $400MHz$. This clock speed is feasible even at the lowest footprints (highest capacitances and thus slowest busses) with sufficient timing margin. The bus length is $10mm$, with repeaters every $2mm$. All capacitances and resistances used in the simulations are taken from a bus lying in *Metal2* above substrate in $0.13\mu m$ CMOS technology.

In Table 2 we present the obtained power figures from both simulations and theoretical estimations. All figures are expressed in EGP (we define one EGP as the dissipation of a 2-input $NAND$ driving two 2-input $NANDs$) in order to quickly see the ratios between bus, encoder and decoder

power dissipation.

From the experiments, we see that the theoretical model for both encoders is in accordance with the numbers that we derived from practical network simulations. For the decoders, the theoretical models do not match. This is because of glitches in the decoder XOR trees, due to differences in the delays of the arriving data signals and parities. It should be noted that these glitches cause not only more (unmodeled) transitions in the XOR trees, but also make the error-correcting circuits be activated for very short periods. This means that, for the Hamming decoder, also the multipoint syndrome decoders and the parity comparison XORs are active. These were not considered in our theoretical modeling. Modeling of these effects is also not that feasible because of the inherent complex and data-dependent behavior of glitches in networks. As for the bus wires, it can be seen that our model correctly predicts the power consumption, being the discrepancy between simulation and modeling less than the 10% for all the three considered cases.

In the previous table we have compared experimental and theoretical results obtained for a bus with only 4 information bits. Since our model has shown a good accuracy in predicting the power consumption of an encoded bus implementing either Hamming code or *Dual Rail* code, it can be employed in order to compare encoded bus with larger original wordsizes.

In Figure 5, we show the values for the modeled power consumption (expressed in EGP) for several values of the wordsize of the original data (k), considering only the bus wires, buffered (a) and unbuffered (b). It can be noticed that, in the buffered case, for $k \geq 13$, the Hamming curve is always under the *Dual Rail* one. This means that, as the information length increases, the increase of power due to the larger number of wires and buffers required by the *Dual Rail* technique is not balanced by the power saved by re-

Table 2. Power consumption according to simulation and theoretical model for Hamming and *Dual Rail* codes at footprints 11, 13 and 16 minimum wire pitches wide (minimum wire pitch = $0.41\mu\text{m}$ in a $0.13\mu\text{m}$ CMOS process).

Footprint	<i>Hamming bus</i>				<i>Dual Rail bus (int. sp)</i>				Difference (DR-H)/DR
	Enc	Bus	Dec	Tot	Enc	Bus	Dec	Tot	
FP11 (sim.)	6.0	331	14.7	352	4.2	373	9.5	387	+10.1%
FP11 (calc)	6.0	321	8	335	4.0	343	6.0	353	+5.2%
FP13 (sim)	6.0	281	13.9	301	4.2	276	8.5	288	-4.2%
FP13 (calc)	6.0	270	8.0	284	4.0	254	6.0	264	-7.4%
FP16 (sim)	5.9	241	13.3	261	4.2	223	6.9	234	-10.2%
FP16 (calc)	6.0	229	8	243	4.0	206	6.0	216	-12.6%

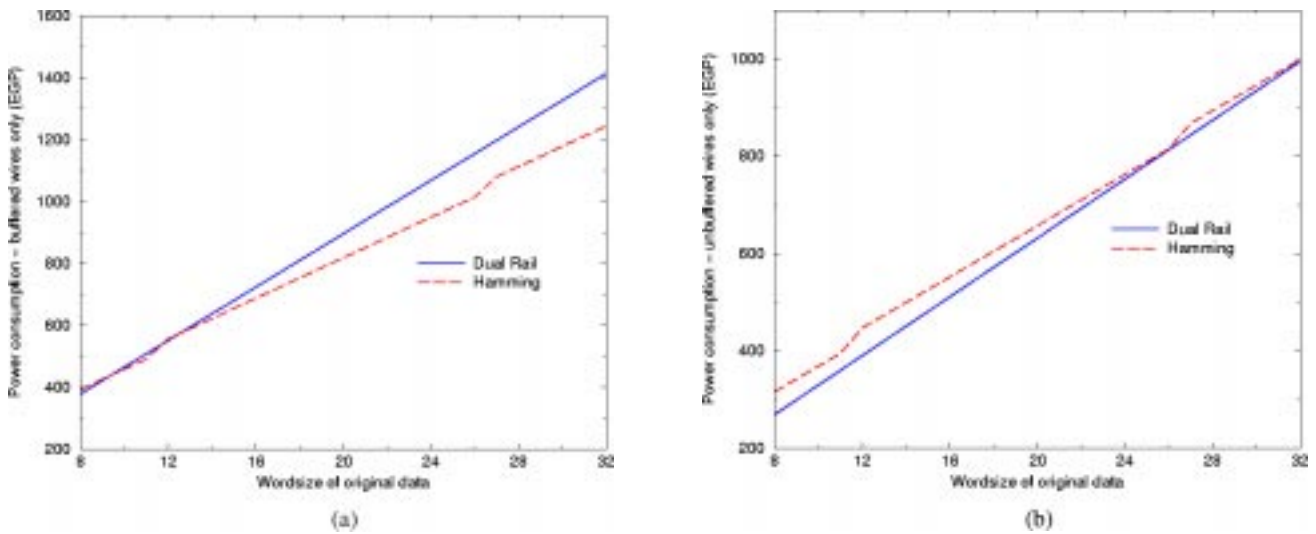


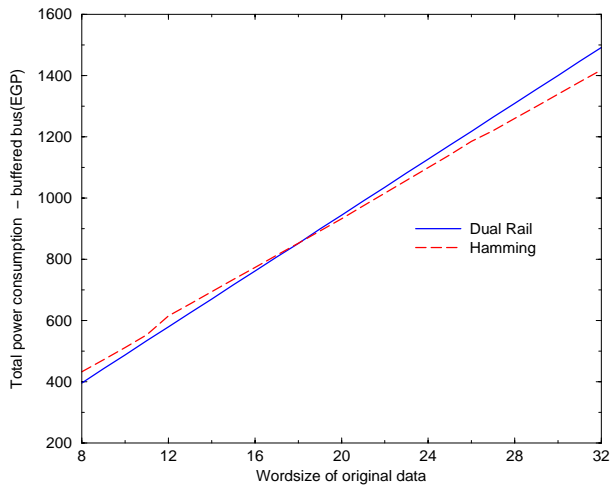
Figure 5. Power consumption of bus wires only: buffered (a) and unbuffered (b).

ducing the number of lateral coupling capacitances to be charged. In this regard, it is worth noticing that the total codeword length n (where $n = k + p$) increases linearly with k in the *Dual Rail*, while for Hamming it increases as $\lceil \log_2 k \rceil$. On the contrary, in the unbuffered case (Figure 5b), the power consumption of the bus for *Dual Rail* is always lower than that for Hamming, for the considered codeword lengths. In this case, the reduction of mutual capacitances to be charged allowed by *Dual Rail* produces a power reduction greater than the extra power required by the larger number of wires, and hence bottom capacitances, with respect to Hamming. As the difference between *Dual Rail* and Hamming decreases with the increase of the wordsize of the original data, we can expect Hamming is preferable for longer codewords.

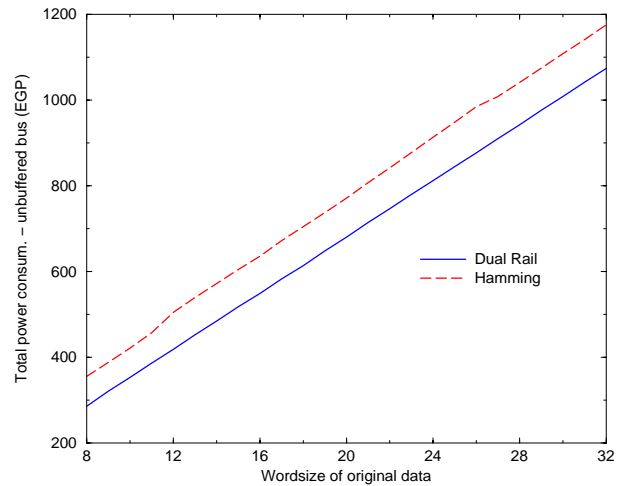
Now, let us analyze the total power consumption considering also the codec circuitry of the two coding techniques, as shown in Figure 6 for buffered (a) and unbuffered (b) bus wires. It can be seen that, in the case with buffered wires, the power dissipation of *Dual Rail* is lower than that of

Hamming for a wordsize of original data (k) lower than 20, with a difference of about 40 *EGP* for $k = 8$ and 12 *EGP* for $k = 16$. It is worth noticing that, considering also the codec circuits, the break-even point between the two coding techniques shift on the right with respect to the case in which only the (buffered) bus wires are considered.

As for the case of unbuffered bus wires, it is worth noticing that *Dual Rail* consumes less than Hamming for all the values of wordsize of original data in the considered range. In this case, in fact, together with the power savings allowed by the *Dual Rail* codec with respect to that of the Hamming, the power reduction within the bus provided by the *Dual Rail* must be taken into account. Consequently, for shorter busses, as they require less buffers, *Dual Rail* becomes more attractive than Hamming. Moreover, if we want to correct more than one error, a segmented bus can be considered. The bus is split up into smaller busses (segments), each one implementing the chosen error correcting code. For instance, if 4 data bit-wide segments are considered, the results shown in Table 2 still hold, and a power sav-



(a)



(b)

Figure 6. Total power consumption with buffered (a) and unbuffered (b) bus wires.

ing more than 10% can be provided by *Dual Rail*, with respect to Hamming, independently of the original buswidth.

6 Conclusion

In this paper we have presented a model developed to predict the power consumption within an encoded bus. In particular, busses implementing both *Dual Rail* and Hamming codes have been considered and compared to each other from the power consumption point of view. Our proposed model has been validated for a small bus by means of electrical simulations. We have then applied the model to cases with larger codeword lengths. We have shown that various tradeoffs can be achieved for different codeword lengths and bus model (buffered or unbuffered). For instance, if a segmented bus is implemented, with each segment 4 or 8 bit-wide, our results predict that *Dual Rail* consumes less power than Hamming at the same footprint, for all the values of the wordsize of the original data.

References

- [1] 2001 International Technology Roadmap for Semiconductors. <http://public.itrs.net/>.
- [2] L. Anghel and M. Nicolaidis. Implementation and Evaluation of a Soft-Error Detecting Technique. In *Proc. of 5th IEEE Int. On-Line Testing Work.*, pages 60 – 65, 1999.
- [3] W.-C. Cheng and M. Pedram. Power-Optimal Encoding for a DRAM Address Bus. *IEEE Trans. on VLSI Systems*, April 2002.
- [4] R. Hegde and N. R. Shanbhag. Toward Achieving Energy Efficiency in Presence of Deep Submicron Noise. *IEEE Trans. on VLSI Systems*, August 2000.
- [5] P. K. Lala. *Self-Checking and Fault-Tolerant Digital Design*. Morgan Kaufmann, 2001.
- [6] C. Metra, M. Favalli, and B. Riccò. On-Line Detection of Logic Errors due to Crosstalk, Delay, and Transient Faults. In *Proc. of IEEE Int. Test Conf.*, pages 524 – 533, 1998.
- [7] C. Metra, M. Favalli, and B. Riccò. Self-checking detection and diagnosis scheme for transient, delay and crosstalk faults affecting bus lines. *IEEE Trans. Comput.*, pages 560 – 574, June 2000.
- [8] M. Nicolaidis. Scaling Deeper to Submicron: On-Line Testing to the Rescue. In *Proc. of IEEE Int'l Test Conference*, 1998.
- [9] M. Nicolaidis. Time Redundancy Based Soft-Error Tolerance to Rescue Nanometer Technologies. In *Proc. of IEEE VLSI Test Symp.*, 1999.
- [10] D. K. Pradhan. *Fault Tolerant Computing: Theory and Techniques*. Prentice Hall, Englewood Cliffs, New Jersey, 1986.
- [11] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj. A Coding Framework for Low-Power Address and Data Busses. *IEEE Trans. on VLSI Systems*, June 1999.
- [12] T. R. N. Rao and E. Fujiwara. *Error Control Coding for Computer Systems*. Prentice Hall, Englewood Cliffs, New Jersey, 1989.
- [13] D. Rossi, V. E. S. van Dijk, R. P. Kleihorst, and A. K. N. an C. Metra. Coding Scheme for Low Power Consumption Fault Tolerant Bus. In *IEEE Proc. of Int. On Line Testing Workshop*, 2002.
- [14] P. P. Sotiriadis and A. Chandrakasan. Bus Energy Minimization by Transition Pattern Coding in Deep Submicron Technologies. In *IEEE/ACM Int. Conference on Computer Aided Design, ICCAD*, pages 322 – 327, 2000.
- [15] P. P. Sotiriadis, A. Wang, and A. Chandrakasan. Transition Pattern Coding: An Approach to Reduce Energy in Interconnect. In *ESSCIRC 2000, Stockholm, Sweden*, 2000.
- [16] M. R. Stan and W. P. Burleson. Low-Power Encodings for Global Communication CMOS VLSI. *IEEE Trans. on VLSI Systems*, December 1997.