

## ADAPTIVE ERROR PROTECTION FOR ENERGY EFFICIENCY

*Lin Li, N. Vijaykrishnan, Mahmut Kandemir, and Mary Jane Irwin*

Microsystems Design Lab  
Pennsylvania State University  
University Park, PA 16802  
{lili, vijay, kandemir, mji}@cse.psu.edu

### ABSTRACT

With dramatic scaling in feature sizes, noise resilience is becoming one of the most important design parameters, similar to performance and energy efficiency. Noise resilience is particularly problematic in long on-chip buses of complex single chip systems such as on-chip multiprocessors. While one might opt to employ a very powerful error protection scheme, this may not be very energy efficient as noise behavior typically varies over time. In this paper, we propose an adaptive error protection scheme for energy efficiency, where the type of the coding scheme is modulated dynamically. The idea behind this strategy is to monitor the dynamic variations in noise behavior and use the least powerful (and hence the most energy efficient) error protection scheme required to maintain the error rates below a pre-set threshold. Our detailed experimental results obtained through simulation show that this adaptive strategy achieves the same level of error protection as the most powerful strategy experimented, without experiencing the latter's energy inefficiency. Based on our results, we recommend system designers to adopt adaptive protection schemes in environments where both energy and reliability are important.

### 1. INTRODUCTION

The continued scaling in process technologies makes it imperative to consider reliability as a first class design constraint. Even the International Technology Roadmap for Semiconductors (ITRS) acknowledges reliability as a cross-cutting problem concerning both designers and test engineers. The reliability of digital circuits is affected by a number of different sources of noise [3, 10, 2, 4], which include the internal noises such as power supply noise, cross talk noise, and inter symbol interference, and the external noises such as electromagnetic interference, thermal noise, slot noise, and noise induced by alpha particle effects. While many of these noise sources have been known to digital designers for a long time, the main challenge is their increased prominence with shrinking feature sizes and the consequent lower supply voltages, smaller nodal capacitances, more closely routed wires and higher clock frequencies.

The need for error protection and recovery mechanisms is becoming an integral part of system design, particularly in complex single chip systems such as on-chip multiprocessors. To guarantee reliability, many error correction codes and error detection codes such as parity, Hamming code, and Berger code are increasingly

used to protect the data stored in memories and data transmitted in the buses. Different coding methods have different capabilities to detect or correct errors induced by different noise sources. While the coding scheme adopted can be designed for the worst case noise scenario, such an approach will be inefficient in terms of both energy and performance.

It must be observed that the induced noise fluctuates due to various environmental factors (such as altitude for soft error rates) and operational conditions (e.g., supply voltage variations due to changes in switching activity). Hence, it is necessary to design a system with self-embedded intelligence to provide the minimum amount of protection to meet the desired reliability levels. This paper focuses on ensuring the reliability of communications in on-chip multiprocessors and concentrates on the data buses connecting the private L1 caches and the shared L2 cache. Specifically, we design and evaluate an adaptive error protection scheme that varies the coding technique based on the dynamic variations in error rates. This scheme helps to reduce the energy consumption by employing a less complex protection mechanism whenever appropriate. Our objective is to maintain the same degree of system reliability provided by the most powerful protection mechanism without always incurring its high energy cost.

In this paper, we make the following contributions:

- We build an architectural level transmission error monitoring system that employs an extra bus line to track the variation in noise sources. This extra line is operated at a lower voltage so that it is more sensitive to the variations in noise behavior.
- Based on the change in noise behavior detected by the monitoring system, the strength of the error protection scheme is suitably adapted. Specifically, we dynamically switch between different error detection schemes to ensure that a predefined reliability constraint is met while consuming the least energy.

We evaluated the effectiveness of our adaptive strategy by using a multiprocessor simulator and two different noise profiles. The results show that the proposed dynamic strategy reduces the bus communication energy consumption by approximately 10% over the most powerful detection scheme (TED in this paper) while providing an error detection rate comparable to that of the latter.

The remainder of this paper is organized as follows. In the next section, we present the noise model and the details of the error detection schemes that we employ. Section 3 explains our adaptive error protection scheme. The experimental framework and the simulation results are provided in Section 4. Next, we discuss related work in Section 5. Section 6 provides conclusions.

---

This work was supported in part by grants from GSRC, NSF Grants 0103583, 0082064 and CAREER Awards 0093082, 0093085.

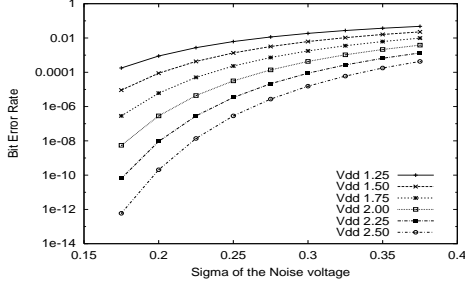


Figure 1: BER with different  $V_{dd}$  and  $\sigma$ .

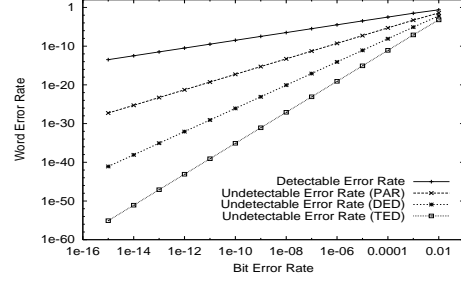


Figure 2: Error rate of different EDC methods.

## 2. PRELIMINARIES

### 2.1. Noise Model

In this work, we are interested in modeling the noise sources influencing the data values in the on-chip interconnects. We adopt the model proposed in [5], that models the different noise sources impacting the bus line as a single Gaussian noise source. This model has also been adopted in other recent efforts for evaluating reliable on-chip interconnects [1, 13]. Specifically, assuming a noise voltage  $V_N$  distributed according to a normal distribution with a variance of  $\sigma^2$  and that a noise voltage greater than  $V_{dd}/2$  causes an output error, the probability of error happening in one transmission line (i.e., one bit of a bus) is denoted as bit error rate (BER)  $\epsilon$  and can be given by

$$\epsilon = Q\left(\frac{V_{dd}}{2\sigma}\right), \quad (1)$$

where  $Q(x)$  is given as  $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy$ .

Note that the probability of error is sensitive to both the supply voltage ( $V_{dd}$ ) and the variance. Specifically, a lower voltage or a larger variance increases the possibility of an error as illustrated in Figure 1, which shows the BER with different  $V_{dd}$  and  $\sigma$ .

In order to model the impact of temporal variations in the noise, we vary  $\sigma$  over the execution of the application. This variation causes a change in distribution of the magnitude of noise voltages generated. Our objective in changing  $\sigma$  is to model physical effects such as variations of flux distribution of alpha particles with altitude/latitude or changes in crosstalk patterns based on data activity in adjacent buses.<sup>1</sup>

Based on this one bit error model, we capture the error in each bit line of the target 32-bit bus as independent and identically distributed random variables (all bit lines use the same  $\sigma$  at any given time). Note that multiple errors occur when two or more individual bit lines generate error at the same cycle.

### 2.2. Error Protection Schemes

In this work, we make use of error-detection schemes coupled with re-transmission on identified errors as the means for error protection. Our choice was motivated by the observation in [1] that error-detection schemes with retransmission are less costly in terms of energy consumption than error-correction schemes at low

Table 1: Power consumption and normalized area (with respect to Encoder for PAR) for different error coding schemes.

	Energy (mW)			Area		
	Encoder	Decoder	Total	Encoder	Decoder	
PAR	4.74	4.92	9.66	100%	1.00	1.03
DED	9.96	11.91	21.88	226%	1.91	2.32
TED	12.58	17.41	29.99	310%	2.43	3.45

error rates typical in on-chip environments. We consider three error detection schemes of different strengths in our evaluation.

- Parity (PAR) uses one extra bit to detect all odd number of bit errors and is a well known representative of single-bit error detection schemes.
- Double Error Detection (DED) uses a (38,32) Hamming code that can detect all single and double bit errors and a subset of multiple bit errors. This scheme uses 6 additional bit lines to carry protection information for the 32-bit bus.
- Triple Error Detection (TED) employs an extra parity bit in addition to the (38,32) Hamming code in order to detect all single, double, and triple bit errors and a subset of higher bit errors [11].

In order to implement these codes, we require an encoder at the sender and a decoder at the receiver of the bus. Consequently, we modeled the encoders and decoders for PAR, DED, and TED in VHDL and synthesized them using a target 0.25um library to obtain their power consumption and layout area characteristics shown in Table 1. While the TED coding scheme provides the best error detection ability, its encoding plus decoding consumes 3.1 (1.4) times the energy required by PAR (DED). Also, there is additional energy consumed in the extra bus lines used in TED and DED as mentioned earlier. The energy consumed in the bus is evaluated using the following equation:

$$E_{bus} = s * C_L * V_{dd}^2 * N_{lines} * N_{trans}, \quad (2)$$

where  $s$  is the average switching activity,  $C_L$  is 5 pF in our evaluation to model an on-chip bus,  $V_{dd}$  is 2.5V,  $N_{lines}$  are 33, 38, 39 for PAR, DED, and TED, respectively. Here,  $N_{trans}$  is the number of transactions (including the retransmissions when errors are detected).

In order to illustrate the differences in the error protection offered by the three schemes considered, Figure 2 shows the resulting word error rate for different bit error rates, and Figure 3 shows the word error rate as a function of varying  $\sigma$ . It is clear that the TED is superior in offering much lower undetectable error rates as compared to DED and PAR. It must be observed that undetectable

<sup>1</sup>An alternate way of modeling such temporal changes in noise would be modulating the number of errors injected in a given interval.

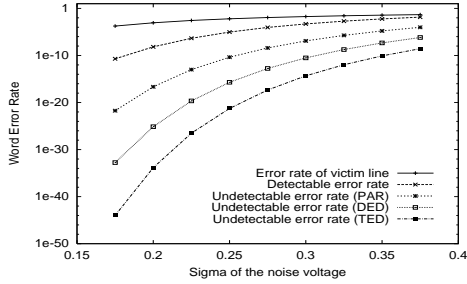


Figure 3: Variation in word error rate as a function of  $\sigma$ .

errors are of serious concern as they can lead to faulty outputs or system crashes. In this figure, the detectable error rates of the different schemes are difficult to distinguish in the scale used and are represented by a single curve (named detectable error rate). This graph also indicates that minor changes in detectable errors magnify to huge variations in undetected errors (The plot for victim line will be explained later). Note that the target reliability metric of undetectable word error rate cannot be observed directly. In order to impose the required bounds on this unmeasurable metric, we utilize its relationship with the detectable error rates in our implementation. From the above discussion, it is clear that there are inherent tradeoffs between error protection capability and energy consumption. Our adaptive strategy to be presented in the next section exploits this tradeoff.

### 3. ADAPTIVE ERROR PROTECTION

Our objective is to adapt the strength of error detection scheme dynamically based on the noise behavior observed. Using only the simplest error detection coding method (PAR), while being most energy-efficient, will lead to high number of undetectable errors. At the other extreme, adopting the worst case approach and employing the most powerful scheme (TED) catches most of the errors but incurs unnecessary energy consumption most of the time due to its conservative nature. Therefore, if we can monitor the change in noise behavior and switch to the least powerful error detection scheme that can maintain the undetected error rates below specified levels, we can minimize the energy consumption while maintaining required protection levels. The main area overhead of our adaptive method is the size of the encoder and decoder due to providing the functions of PAR, DED, and TED. Synthesis result shows that the total area of encoder and decoder in our method is 29% more than that in TED method.

There are two important aspects of our implementation: (a) detecting the variation in noise behavior and (b) identifying the protection scheme to employ for the observed noise behavior. An indicator of the variation in the noise behavior is the detected error rates observed at the decoders of the bus. When detectable error rate is small, it requires a long period before the number of detectable errors change and this makes it difficult to use the counted errors to effect changes in the protection scheme promptly. For example, when  $\sigma$  is 0.3, the detectable error rates of PAR, DED, and TED are  $5.09E-4$ ,  $5.87E-4$ , and  $6.02E-4$ , respectively, and they result in only five or six errors in 10,000 transmissions. Coupled with the observation that even small variations in detectable error rates indicate huge variations in undetectable error rates, it becomes im-

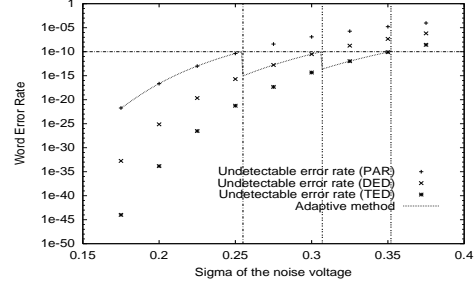


Figure 4: Word error rate in our adaptive method as a function of  $\sigma$ .

Table 2: Influence of sampling window size on the number of errors detected.

window size	$\sigma = 0.255$		$\sigma = 0.307$		$\sigma = 0.352$	
	Min	Max	Min	Max	Min	Max
1,000	0	21	4	42	15	68
10,000	41	106	153	266	312	461
100,000	620	822	1939	2241	3595	4011
1,000,000	6858	7360	20499	21292	37441	38303

portant to devise an alternate monitoring scheme. For this purpose we utilize a *victim bus line* that amplifies the number of detectable errors in order to detect changes in the noise behavior quickly. This victim line uses half the voltage swing as the normal bus lines and is more susceptible to variations in noise. Consequently, we can observe from Figures 3 that the detectable error rate of the victim line is much higher than those of PAR, DED, and TED schemes. In our implementation, the victim line transmits a repeated sequence of 0-1 with each data transmission and one per 10 cycles when the bus is in idle state. Whenever, the decoder of this line receives the same bit value in two successive transmissions it indicates a detected error.

The main goal of our adaptive approach is to keep the undetectable word error rate below a preset value (threshold) when the noise variable  $\sigma$  changes. For purpose of evaluation, we set the target undetectable word error rate to  $1E-10$ . When more than one coding scheme can achieve the target undetectable word error rate, we choose the scheme that has the lowest energy consumption. Figure 4 shows the behavior of our adaptive strategy as  $\sigma$  changes. Specifically, when  $\sigma$  increases, the simple PAR method is used at first. When the undetectable word error rate of PAR exceeds the predefined value  $1E-10$ , our method switches to DED, and later to TED when DED can no longer provide the required reliability. As a result, one can identify three thresholds,  $\sigma = 0.255$ ,  $0.307$ , and  $0.352$  for switching between the different schemes from this figure. It must be noted that the strength of coding can also be reduced using the same thresholds. Further, when  $\sigma$  is over  $0.352$ , our implementation cannot guarantee the required reliability level. In order to identify these switching points of  $\sigma$ , we utilize two counters, a saturating counter to track a sampling window and another counter to maintain the total number of detected errors on the victim line during this window. At the end of every sampling window, the error counter is reset.

In order to choose an appropriate sampling window size, we simulate 10 million transmissions with different  $\sigma$  values associated with the three thresholds mentioned above and experiment with different window sizes to count the number of errors on the

Table 3: The number of transactions and execution cycles for the Splash2 benchmarks.

	transactions	cycles
barnes	10098956	490339742
ocean1	35087162	1719408062
ocean2	22634451	2039085568
radix	7776529	722890419
raytrace	22299175	1108642808
water1	5468576	442982664
water2	9083072	361669709

victim line. Table 2 shows the minimum and maximum values of errors detected in the simulation across all the sample window sizes considered. If a small window size is selected, the system can react faster to changes in noise. However, a very small window size such as 1,000 transmissions cannot distinguish between different noise levels due to the overlap between the maximum and minimum errors that can be observed. In contrast, a large value for sampling window size can clearly distinguish between the different noise levels but will react too slowly to the changes in noise behavior. Based on the results in Table 2, we choose a sampling window size of 10,000 transmissions for our experiments.

Having selected the sampling window size, we next determine the thresholds with which to compare the error counter values for switching between different coding schemes. Using equation 1 and the three  $\sigma$  values corresponding to the thresholds, we compute the BER. Then utilizing the BER and assuming the maximum number of possible transmissions that can occur within our given sample window size, we compute the number of errors corresponding to the three cases as 71, 208 and 379. In order to avoid thrashing between the choices of the schemes when noise behavior straddles across the boundaries, we set the following rules for switching: switch from PAR to DED when  $\sigma$  exceeds 0.255 (counter value 71) and is below 0.307 (counter value 208); switch from PAR/DED to TED when  $\sigma$  exceeds 0.307 (counter value 208); switch from TED/DED to PAR when  $\sigma$  is below 0.235 (counter value 39); from TED to DED when it is below 0.287 (counter value 147) and higher than 0.235 (counter value 39). It is essential for both the encoders and decoders to be notified of the change required in the coding scheme. In our implementation, this task is performed by transmitting a special code to all encoders and decoders using the data bus. This transmission occurs quite infrequently and is observed to cause a negligible impact on performance and energy.

## 4. EXPERIMENTAL EVALUATION

### 4.1. Simulation Environment

Our target system is an on-chip multiprocessor where each processor has private L1 data and instruction caches and the processors share an on-chip L2 cache. To model this system, we use MP\_simplescalar [8], a multiprocessor version of the Simplescalar simulator. The default configuration contains four processors and has L1 (8K) and L2 (256K) access latencies of 1 cycle and 10 cycles, respectively. The main memory latency is 100 cycles. The focus of the adaptive strategy is the data bus between the L1 caches and the L2 cache, which is the longest on-chip interconnect in our design other than the clocking network. It should be noted that our approach can also be applied to other buses and architectures.

We evaluate our strategy using the benchmarks from Splash2, a widely used parallel benchmark suite [12]. Each benchmark is

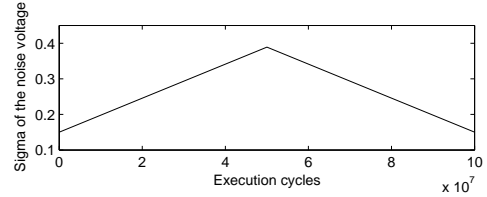


Figure 5: Noise profile 1.

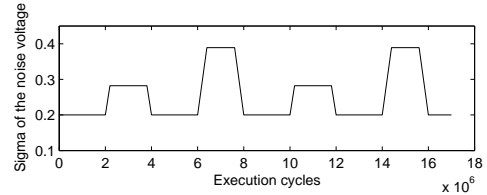


Figure 6: Noise profile 2.

simulated for 500 million instructions and the corresponding number of transactions and execution cycles are given in Table 3. These values correspond to the case when no error protection is implemented.

In order to simulate the noise activity, we use two different types of noise profiles. The first profile mimics a scenario where there is a repeated pattern of a slow increase in the severity of the noise problem followed by a phase of decrease. To model this,  $\sigma$  is changed from 0.150 to 0.389 continuously in the first part and from 0.389 to 0.150 in the second part of a period of 100 million cycles (this profile repeats itself throughout execution) as shown in Figure 5. The second profile corresponds to the case where there are more frequent and random changes in the noise behavior and is modeled as illustrated in Figure 6. The noise in this profile repeats every 8,000,000 cycles. Here,  $\sigma$  is 0.2 by default and varies to 0.282 from cycle 2,000,000 to 4,000,000 and to 0.389 from cycle 6,000,000 to 8,000,000.

### 4.2. Results

In this section, we present the energy and reliability impact of our adaptive error protection strategy. First, we focus on the energy behavior and give in Figures 7 and 8 the energy consumption for noise profiles 1 and 2, respectively. In these figures, the energy consumption is broken down (from bottom to top in each bar) as that consumed by the 32-bit data transfer (bus line), the energy consumed in the additional bit lines for supporting the coding scheme (code lines), the energy consumed by the encoders and decoders (coder) and the energy consumed by the victim line in the case of our adaptive approach (victim). The energy for retransmissions is included. It can be observed that in all the coding strategies the bulk of the energy is consumed by the data transfers and that the additional overhead for error protection ranges from 12.7% for PAR to 51.8% for TED.

The adaptive approach provides an average energy saving of 8% (12%) over the TED approach for noise profile 1 (noise profile 2) when both of them provide our target undetectable error rate. In addition, the adaptive approach consumes 1% (5%) less energy using noise profile 1 (noise profile 2) even as compared to the DED scheme while providing better resilience to errors.

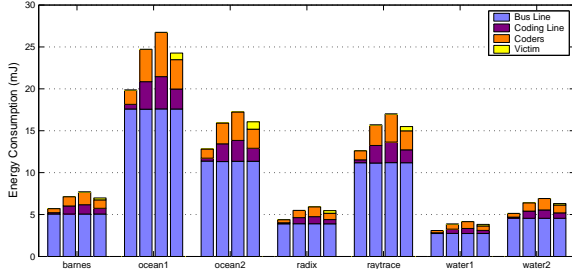


Figure 7: Energy behavior with noise profile 1. The four bars for each benchmark from left to right correspond to using PAR, DED, TED and our adaptive approach.

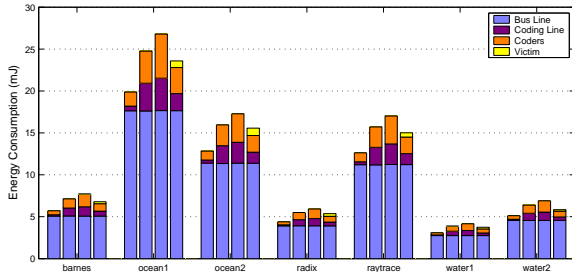


Figure 8: Energy behavior with noise profile 2. The four bars for each benchmark from left to right correspond to using PAR, DED, TED and our adaptive approach.

It should be emphasized that the relative gains of our scheme are based on the relative energy overhead of the coding schemes as compared to the energy consumed by the actual data transfers. For example, when error protection is required for a bus with half the capacitive load (2.5pF per bit line) considered in results shown in Figure 7, energy saving of the adaptive approach over TED increases to 13% (from 8%). In order to better understand the source of our energy savings over TED, we give in Figures 9 and 10 the cumulative cycles spent in each error protection scheme (PAR, DED and TED) when the adaptive strategy is used. It can be observed that the adaptive strategy makes use of the PAR and DED schemes that have a smaller energy overhead around 60% (80%) of the time when using noise profile 1 (noise profile 2).

While in all our experiments the adaptive strategy generated a better energy behavior than TED, some anomalous cases can result in our strategy consuming more energy than TED. For example, if the noise level is continuously severe requiring TED always, the overhead of the victim line makes the adaptive strategy more

Table 4: Error behavior when using noise profile 1. (D: Detected Error, U: Undetected Error)

	PAR		DED		TED		ADAPTIVE	
	D	U	D	U	D	U	D	U
barnes	25652	124	29590	1	30446	0	30435	0
ocean1	83346	461	96359	1	98876	0	98845	0
ocean2	57422	346	66183	0	67956	0	67927	0
radix	17293	97	19927	0	20446	0	20442	0
raytrace	57551	331	66547	1	68344	0	68318	0
water1	11346	53	13105	0	13435	0	13428	0
water2	20090	108	23175	0	23798	0	23785	0

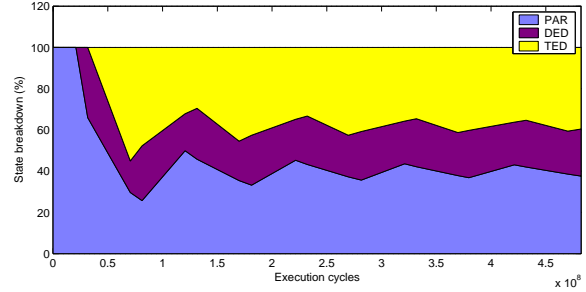


Figure 9: State breakdown based on coding schemes used (noise profile 1).

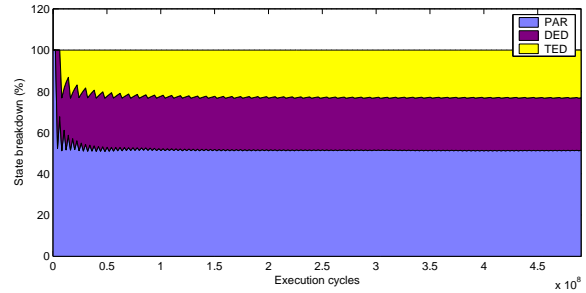


Figure 10: State breakdown based on coding schemes used (noise profile 2).

energy consuming than TED. As another example, if the number of transmissions is very low, the energy overhead of the victim line (that transitions every 10 cycles) can become a bottleneck. However, such cases did not happen in the realistic workloads we experimented with.

Next, we focus on the error behavior of the different encoding strategies. Tables 4 and 5 show the number of detected and undetected errors for noise profiles 1 and 2, respectively. It must be observed that our adaptive scheme is as powerful as the TED approach and leaves no errors undetected. In contrast, the DED approach leads to 3 (6) undetected errors using noise profile 1 (noise profile 2). Furthermore, we see that PAR is clearly insufficient for the noise profiles used in this experimentation. Also, it must be observed that the number of actual errors that happen during transmission is different for the different schemes as DED and TED use additional bus lines for coding as compared to PAR. As a consequence of the additional lines, the number of errors increases.

Table 5: Error behavior when using noise profile 2. (D: Detected Error, U: Undetected Error)

	PAR		DED		TED		ADAPTIVE	
	D	U	D	U	D	U	D	U
barnes	48768	476	56597	0	58066	0	57908	0
ocean1	171724	1732	199050	1	204323	0	203754	0
ocean2	109998	1056	127404	0	130759	0	130358	0
radix	37302	356	43219	1	44410	0	44273	0
raytrace	108992	1057	126442	2	129807	0	129488	0
water1	26163	250	30283	2	31108	0	31012	0
water2	30901	303	35681	0	36692	0	36686	0

## 5. RELATED WORK

There have been several efforts focusing on the energy optimization of complete systems and the interconnect in particular. Several coding and adaptive coding schemes have been proposed in literature for reducing power consumption. Since our work focuses on providing reliable communication, we limit our discussion to those techniques that consider energy and reliability together.

A good overview of the noise problems in submicron CMOS is presented in [9]. In addition, this work presents a noise tolerance scheme for achieving energy and performance efficiency in the presence of noise. In [5], the authors present lower bounds on energy consumption in the presence of noise and show the effectiveness of coding schemes such as Hamming codes in reducing the energy consumption when communicating in noisy buses. Hedge and Shanbhag [6] explore the possibility of handling errors by compensation through DSP algorithms. Bertozzi et. al. [1], compare the energy behavior of different error correction and detection schemes. Their results show that error detection methods combined with retransmission are more energy efficient than error correction methods under the same reliability constraints. Therefore, in our work, we select different error detection methods which have different ability as the candidates of our adaptive approach.

Most closely related to our work is the approach to scaling the supply voltage based on the observed error pattern [13]. In contrast to our approach that dynamically changes the coding strength, the scheme in [13] exploits the fact that a smaller voltage (hence, a smaller noise margin) would be sufficient for transmission in a less noisy phase of execution. Therefore, they increase (decrease) the voltage when the noise increases above (decreases below) a threshold point. Our initial evaluation shows that changing the coding scheme strength while retaining the same  $V_{dd}$  can provide a larger improvement in reliability per unit increase in energy consumption as compared to increasing the supply voltage using the same coding method. To quantify this, we present example results from experiments using 1 million transactions on Barnes application using a constant  $\sigma$  of 0.3. Using PAR with a supply voltage of 2V provides an undetectable error rate of  $9.59E-5$ . When we employ the more powerful TED approach using the same 2V supply voltage, the undetectable error rate is reduced to  $2.75E-9$ . The TED approach consumes 1.18mJ more energy than PAR for providing the additional reliability. In comparison, the voltage scaling approach needs the voltage of PAR to be increased from 2.0V to 2.75V to reduce undetectable error rate to  $2.75E-9$  and consumes an additional 3.16mJ of energy (as compared to PAR at 2.0V) for the additional reliability. These results show the usefulness of the adaptive coding schemes even in the presence of voltage scaling. However, it is more beneficial to consider the two approaches as complementary and devise schemes that combine voltage scaling and coding adaptation. Such a combination is particularly relevant given that number of different coding schemes that can be implemented simultaneously and the number of supply voltages/supply scaling range are typically limited in real systems. Investigation of the combination of these schemes will be the focus of our future work.

In contrast to our focus on providing energy-aware noise protection schemes, there have been several techniques that attempt to reduce the energy consumption resulting in buses due to the presence of noise. An example of such optimizations is the work by Kim et. al. [7] that optimizes the coupling power consumed by

interconnects due to the presence of cross talk noise.

## 6. CONCLUSIONS

Noise immunity is becoming a major problem with the scaling process technologies. Unfortunately, adopting the most comprehensive protection mechanism may not be very desirable from energy and performance perspectives. This paper proposes and evaluates an adaptive error protection scheme where the strength of the error coding scheme is varied to trade off between energy consumption and error protection. In order to test the effectiveness of this strategy, we performed simulation of an on-chip multiprocessor interconnect with different noise profiles. Our results indicate that the proposed strategy achieves the same level of reliability as TED while reducing the energy consumption of the latter. Also, we observed that it outperforms the DED scheme from both energy and reliability angles.

## 7. REFERENCES

- [1] D. Bertozzi, L. Benini, and G. de Micheli. Low power error resilient encoding for on-chip data buses. In *DATE 2002*, pages 102–109, 2002.
- [2] H. H. Chen and D. D. Ling. Power supply noise analysis methodology for deep-submicron vlsi chip design. In *34th DAC*, pages 638–643, 1997.
- [3] M. CuvIELLO, S. Dey, X. Bai, and Y. Zhao. Fault modeling and simulation for crosstalk in system-on-chip interconnects. In *IEEE/ACM ICCAD 1999*, pages 297–303, 1999.
- [4] W. J. Dally and J. W. Poulton. *Digital systems engineering*. Cambridge University Press, 1998.
- [5] R. Hegde and N. R. Shanbhag. Toward achieving energy efficiency in presence of deep submicron noise. *IEEE Transactions on VLSI Systems*, 8(4):379–391, 2000.
- [6] R. Hegde and N. R. Shanbhag. Soft digital signal processing. *IEEE Transactions on VLSI Systems*, 9(6):813–823, 2001.
- [7] K.-W. Kim, S.-O. Jung, U. Narayanan, C. L. Liu, and S.-M. Kang. Noise-aware power optimization for on-chip interconnect. In *ISLPED 2000*, pages 108–113, 2000.
- [8] N. Manjikian. Multiprocessor enhancements of the simulators tool set. *ACM SIGARCH Computer Architecture News*, 29(1):8–15, 2001.
- [9] N. Shanbhag, K. Soumyanath, and S. Martin. Reliable low-power design in the presence of deep submicron noise (embedded tutorial session). In *ISLPED 2000*, pages 295–302, 2000.
- [10] K. L. Shepard and V. Narayanan. Noise in deep submicron digital design. In *IEEE/ACM ICCAD 1996*, pages 524–531, 1996.
- [11] J. F. Wakerly. *Digital design: principles and practices*. Prentice-Hall, 2000.
- [12] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta. The splash-2 programs: characterization and methodological considerations. In *22nd ISCA*, pages 24–36, 1995.
- [13] F. Worm, P. Ienne, P. Thiran, and G. D. Micheli. An adaptive low-power transmission scheme for on-chip networks. In *15th ISSS*, pages 92–100, 2002.