Performability/Energy Trade-off in Error-Control Schemes for On-Chip Networks

Alireza Ejlali, Bashir M. Al-Hashimi, *Senior Member, IEEE*, Paul Rosinger,
Seyed Ghassem Miremadi, *Member, IEEE*, and Luca Benini, *Fellow, IEEE*

***Abstract*—High reliability against noise, high performance, and low energy consumption are key objectives in the design of on-chip networks. Recently some researchers have considered the impact of various error-control schemes on these objectives and on the trade-off between them. In all these works performance and reliability are measured separately. However, we will argue in this paper that the use of error-control schemes in on-chip networks results in *degradable systems*, hence performance and reliability must be measured jointly using a unified measure, i.e., *performability*. Based on the traditional concept of performability, we provide a definition for the 'Interconnect Performability'. Analytical models are developed for interconnect performability and expected energy consumption. A detailed comparative analysis of the error-control schemes using the performability analytical models and SPICE simulations is provided taking into consideration voltage swing variations (used to reduce interconnect energy consumption) and variations in wire length. Furthermore, the impact of noise power and time constraint on the effectiveness of error-control schemes are analyzed.**

***Index Terms*—On-chip network, on-chip interconnect, energy consumption, error control, performability**

I. Introduction

The implementation of an on-chip network affects the system *reliability, performance*, and *energy consumption* to a large extent [1]. Energy consumption is one of the most prominent issues in on-chip networks. It has been shown that on-chip interconnects account for a significant fraction of the total on-chip energy consumption [3]. On the other hand, the required reliability of on-chip interconnects is becoming harder to achieve due to shrinking feature-sizes and supply voltage scaling [2].

To address the energy consumption issue, reduced voltage swing [3], [4] is often used. However, reduced voltage swing leads to decreased noise margin; making interconnects less immune to noise. Variations in voltage swing also necessitate changes in interconnect operational frequency which lead to variations in performance [3]. To address the reliability issue, error-control schemes such as Automatic Repeat Request (ARQ), and Forward Error Control (FEC) can be used [2], [3]. However, these mechanisms increase the energy consumption and can degrade the performance of the on-chip networks. For instance, in the ARQ scheme, the receiver requests the sender to retransmit the data unit that was faulty [2]. Clearly, retransmissions take time (i.e., degraded performance) and consume energy (i.e., increased energy consumption). Based on the above, high performance, high reliability and low energy consumption are conflicting objectives that require to be considered jointly when designing an on-chip network.

In the context of on-chip communication, the energy efficiency of FEC and ARQ has been studied in [2]. This research has reported that, for the same constraint on system reliability, ARQ consumes less energy than FEC. However, this research has not considered the performance. Indeed, it has been assumed that timing penalties can be tolerated [8]. Furthermore, this research has not considered the hybrid ARQ/FEC (HARQ) scheme. A dynamic voltage swing approach has been proposed in [3] to optimize the energy consumption of ARQ without degrading the performance and the reliability. However, this research has not considered FEC and HARQ. [9] has compared ARQ and HARQ. This work provides useful information to select an appropriate error-control scheme for a given application. However, it addresses energy/reliability and performance/reliability trade-offs separately and does not consider the impact of voltage swing on the simultaneous trade-off between reliability, performance, and energy consumption. [15], [16], [21] have addressed the reliability, performance and energy consumption of NoCs, however these works are mainly focused on router architecture and they do not investigate the issues related to channel wires such as voltage swing variations, variations in wire length, etc. These works also do not provide any comparison between ARQ, FEC and HARQ.

Manuscript received August 13, 2007; revised January 10, 2008, April 8, 2008 and May 20, 2008. This work was supported in part by the U.K. Engineering and Physical Sciences Research Council (EPSRC), under grants EP/C512804/1 and EP/035965/1, and in part by the research vice-presidency of Sharif University of Technology.

A. Ejlali and S. G. Miremadi are with the Department of Computer Engineering, Sharif University of Technology, Tehran 14588, Iran (e-mail: ejlali@sharif.edu; miremadi@sharif.edu).

B. M. Al-Hashimi is with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: bmah@ecs.soton.ac.uk).

P. Rosinger was with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. He is now with Winton Capital Management, London W8 6LS, U.K. (e-mail: paul.rosinger@zepler.org).

L. Benini is with the University of Bologna, 40136 Bologna, Italy (e-mail: lbenini@deis.unibo.it).

Although some of the above previous works have addressed the performance and reliability of NoCs, none of them has addressed the performability metric [12] which is a composite measure of performance and reliability. It has been shown that for degradable fault tolerant systems – fault tolerant systems that tolerate faults by reducing their performance – reliability and performance cannot be measured separately and should be measured jointly using the performability metric [12]. We will argue in this paper (Section II-B) that the use of error-control schemes in on-chip networks results in degradable fault tolerant systems, hence performability should be used to measure performance and reliability jointly. Based on the traditional concept of performability [5], [12], in this paper, we provide a definition of "interconnect performability" to measure the reliability and performance of an on-chip network interconnect in a composite way. Two other important issues which have not been addressed in all previous works are the impacts of (i) time constraints and (ii) noise power on the effectiveness of error-control schemes. In this paper, we aim: (i) to analyze the impact of voltage swing and different error-control schemes on the trade-off between performability and energy, and (ii) to answer the following question: "If a message transmission has to be finished in a given time interval (time constraint) and in the presence of noise with a given power, which error-control scheme and what voltage swing must be used to perform the transmission with the minimum energy and highest performability?".

To analyze the performability/energy trade-off, analytical models of performability and expected energy consumption are developed for three error-control schemes (ARQ, FEC, and HARQ) and the simple non-fault-tolerant communication (SNFT). In the energy analysis, the energy overhead of the error-control circuits, estimated by SPICE simulations, is also considered. We have chosen SNFT to demonstrate why error-control schemes are necessary.

The rest of the paper is organized as follows. Section II provides the performability/energy models for communication schemes. Based on the models provided in Section II, Section III analyzes and compares the different communication schemes. Finally, Section IV concludes the paper.

II. Error-control schemes and performability/energy models

One of the distinctive aspects of on-chip networks is data packetization [1]. In general, each message to be transmitted is partitioned into packets. Packets in turn are often broken into message flow-control units or *flits*. Most of the related works [3], [7], [8] consider flit-level error control where each flit contains its own check bits. Similarly, in this paper we consider flit-level error control. Fig. 1 shows a possible architecture for an on-chip interconnect with flit-level error control. The encoder (denoted by 'ENC') adds check bits to each flit and the decoder (denoted by 'DEC') uses the check bits to detect and/or correct faulty flits. The 1-bit connection line denoted by 'Retransmission Request' is, unlike all the other connections in Fig. 1, backward from the decoder to the encoder. The 'Retransmission Request' line is only required for the error-control schemes with retransmission capability and is not required for the other schemes (Section II-A). The level shifter units are used to change the voltage swing.

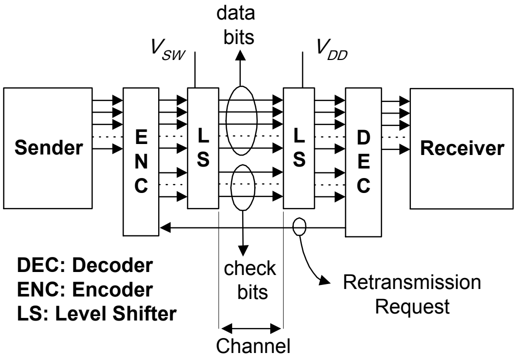


Fig. 1. A possible architecture for an on-chip interconnect

In the rest of this section, we first introduce the error-control schemes, and then we develop the analytical models of performability and energy for the schemes.

A. Error-control schemes

The three error-control schemes for on-chip networks, considered in this work, are:

1) ARQ: In this scheme [2], the sender includes an encoder which encodes flits using an error detection code (e.g., CRC-8 code [3]). The receiver includes a decoder which can detect errors (faulty flits). When the receiver detects no fault in a flit, it sends back an ACK (e.g., a '0' on the 1-bit 'Retransmission Request' line) to the sender to acknowledge the correctness of the flit. However, when the receiver detects that a flit is faulty, it sends back a NACK (e.g., a '1' on the 'Retransmission Request' line) to request the sender to resend the flit. This process is repeated until the receiver detects no fault in the flit. When the receiver detects no fault in a flit, the flit is supposed to be correct; however there are rare occasions when a flit is faulty and the receiver cannot detect the fault. In this case, since the fault is undetected, the receiver does not request a retransmission. Therefore, the flit remains faulty and the transmission fails.

Most of the related works (e.g., [3]) consider the ARQ schemes which are based on a policy called Go-Back-*N* [18]. In this policy, flits are transmitted continuously and the sender does not wait for an ACK after sending a flit. Such an ACK is received after a round-trip delay. The sender requires buffering resources to store a copy of those flits that are transmitted during the round trip delay and their ACKs are still not received. Using these buffers, when a NACK is received, the sender backs up to the flit that is negatively acknowledged and resends it in addition to the *N*-1 (*N* is called window size [18]) succeeding flits that were transmitted during the round-trip delay. A flit is removed from the sender buffer only when an ACK is received for it. At the receiver, the *N*-1 received flits following a detected faulty flit are discarded regardless of whether they were correct or not. It should be noted that in the Go-Back-*N* policy, the channel and the 'Retransmission Request' line operate in parallel. That is, while the sender is transmitting the *i*th flit over the channel, the receiver transmits an ACK/NACK for the *i*-(*N*-1)th flit over the 'Retransmission Request' line. In this paper, we consider the ARQ schemes which are based on the Go-Back-*N* policy (for more information on the Go-Back-*N* policy refer to [18]).

As it can be seen from Fig. 1, the 'Retransmission Request' line is not driven with a reduced voltage swing. This is because this line usually carries ACKs and it rarely carries a NACK, only when a fault is detected. Hence the switching activity of this line is essentially very low, so that a reduced voltage swing is not required.

2) FEC: In this scheme [2], the sender includes an encoder that encodes flits using an error correction code which can be used for single-bit error correction (e.g., overlapping parity bits [6]). The receiver includes a decoder which can correct single-bit errors. When the receiver detects a single-bit error in a flit, it corrects the error. However, on the occasions that there is a multiple-bit error in a flit, it cannot be corrected and the transmission fails. In this scheme, the 'Retransmission Request' line shown in Fig. 1 is not needed and does not exist.

3) Hybrid FEC/ARQ (HARQ): In this scheme, the sender includes an encoder that encodes flits using an error correction code (e.g., overlapping parity bits [2]). The receiver includes a decoder which can correct single-bit errors and detect multiple-bit errors. When the receiver detects a single-bit error in a flit, it corrects the error. However, when the receiver detects a multiple-bit error in a single flit, it cannot correct the error and hence requests the sender, through the 'Retransmission Request' line (Fig. 1), to resend the flit. This process is repeated until the receiver detects no fault in the flit or detects only a single-bit error that is correctable without requiring any retransmission. Like in ARQ, when the receiver detects no fault in a flit, the flit is supposed to be correct; however there are rare occasions when a flit is faulty and the receiver cannot detect the fault. Since the fault is undetected, the receiver neither corrects the flit nor requests a retransmission, therefore the transmission fails. In this paper, the retransmission policy of HARQ is considered to be Go-Back-*N*.

B. Performability of an on-chip network interconnect

An important class of fault tolerant systems are degradable systems which in the presence of faults descend into a lower level of performance but still operate correctly. In fact, degradable systems have the capability of compromising performance for reliability. These are unlike non-degradable fault tolerant systems which in the presence of a fault either tolerate the fault and continue to operate correctly at the normal performance level (without any degradation in performance) or do not tolerate the fault and fail. As discussed in the literature (e.g., [5], [12]), traditional views of computer "performance" and computer "reliability" are no longer applicable to degradable systems and performance and reliability must be measured jointly using a metric called performability. We believe that the use of error-control schemes for on-chip network interconnects may result in degradable systems, thereby requiring performability analysis. We clarify this by means of the following example:

Suppose a 32-bit on-chip interconnect operates at the frequency of 500MHz (i.e., each flit takes 2ns to be transferred and the bit rate is 32bits/2ns = 16Gbit/s) and we want to transfer 10 flits on this interconnect. Also suppose that ARQ is used for this interconnect. If no fault occurs during the transfer of the 10 flits, the transfer of the 10 flits will take 20ns and hence the useful bit rate will be (32*10bits/20ns)=16Gbit/s. However, if for example during the transfer of the 10 flits, 4 of them become faulty and require retransmissions, 14 flits should be totally transferred that will take 28ns and hence the useful bit rate will be (32*10bits/28ns) ≅11.4Gbit/s. It can be seen that when faults have occurred during the transmission of the 10 flits, the faults have been tolerated using ARQ, but the interconnect performance has dropped from 16Gbit/s to 11.4Gbit/s. This example shows that the use of ARQ for the interconnect results in a degradable system. Therefore a performability analysis should be used for such an interconnect rather than analyzing the performance and reliability separately. In fact when we use error-control schemes for on-chip network interconnects, the traditional views of communication performance and communication reliability have the following drawbacks:

1- Metrics such as bit rate, baud rate, latency, bandwidth, and operational frequency are some of the most commonly used measures of communication performance [3], [9]. However, when error-control schemes are used in on-chip networks, these metrics cannot provide a realistic view of performance. In fact, from a performance point of view, it is the useful bit rate which is important, not the apparent rate at which all the bits (including faulty and fault-free flits) are transferred. On the other hand, the use of error-control schemes causes the useful bit rate to become dependent on how faults occur and how they are tolerated. Therefore it may be impossible to measure the real performance without considering the reliability issues. For instance, in the above example, when there is no faulty flit, the useful bit rate is 16Gbit/s, but when 4 flits become faulty, the useful bit rate is reduced to 11.4Gbit/s (although the faults are tolerated). Note that while the useful bit rate varies with the number of faults, the apparent bit rate is constant and equal to 16Gbit/s.

2- Another important drawback of the above mentioned metrics of communication performance is that they cannot model the probabilistic nature of the performance of those on-chip interconnects which use error-control schemes. From the above example, it is clear that the real performance of the example interconnect (i.e., the useful bit rate) depends on the number of faulty flits. However, since faults occur randomly the real performance is also a random variable and is not deterministic. In such cases, metrics such as bit rate, baud rate, etc. can only be used to describe the average (or the maximum) value of the interconnect performance but cannot model its probabilistic nature.

3- Metrics such as Bit Error Rate, Flit Error Rate and Residual Error Probability are some of the most commonly used measures of communication reliability [2], [3], [9]. However, when error-control schemes are used in on-chip networks, these metrics cannot provide a realistic view of how reliable an on-chip interconnect is. For example, suppose that in the above example the residual error probability is 0. From a reliability point of view this is the highest imaginable reliability which means that all the possible faults are definitely detected and tolerated by retransmission. However, if the number of faulty flits increases, although all of them will be detected and tolerated, the interconnect performance may be drastically reduced because of the time that retransmissions will take. In this case, the reliability of the interconnect is apparently infinite since all the faults are tolerated, but the resulting performance reduction may make the interconnect completely useless if the performance becomes less than what is required by the application. Hence, for those on-chip network interconnects that use error-control schemes, performance have to be taken into account in measuring reliability.

The above discussion indicates that like all other degradable systems, when error-control schemes are used for on-chip interconnects, performance and reliability may be impossible to be measured separately and preferably they should be measured jointly using the performability metric. Formal definitions for performability have been provided in [5], [6], [12]. However, the performability of a degradable system can be simply defined as [5]: "the probability of completing a given amount of useful work within a specified time interval". Since in an on-chip network interconnect the useful work is to transmit useful bits (by useful bits we mean original data bits excluding check bits and redundantly transmitted data bits), in this paper we define the performability *P(L,T)* of an on-chip network interconnect as the probability to transmit *L* useful bits during the time interval *T* in the presence of noise. To see how this definition can be used to combine the reliability and performance analysis, again consider ARQ. The presence of faulty flits (low reliability problem) in ARQ necessitates a more frequent retransmission of flits which requires more time and reduces the probability to finish the transmission of a fixed number of useful bits during a fixed time interval (i.e., performability). Also, reducing the bit rate (i.e., low performance problem) increases the time required for sending the flits. This time increase reduces the probability to finish the transmission of a fixed number of useful bits during a fixed time interval (i.e., performability). Whilst the performability of an on-chip interconnect provides a better insight into the performance and reliability of the interconnect, it is not intended to replace the basic metrics of performance and reliability (e.g., Bit Error Rate and operational frequency) with the performability metric. In fact, as it will be seen in this section, the performability metric itself should be calculated and obtained from the basic metrics of performance and reliability.

The analytical performability models for the communication schemes are presented next.

Analytical Performability Models

An effective method to reduce the energy consumption of an on-chip interconnect is to reduce the voltage swing [3], [4]. Variations in the voltage swing of a channel also lead to variations in the channel delay [3]. When a channel is used at the voltage swing V_{SW}, the channel delay is [3]:

$$D_{channel}(V_{SW}) = \frac{C_L}{K_m} . \frac{V_{SW}}{(V_{SW} - V_{th})^2} \tag{1}$$

where K_m is the driver transistor transconductance, C_L is the wire capacitance, and V_{th} is the threshold voltage of the transistors. Let $D_{Error\text{-}control}$ be the additional delay imposed by the error-control circuit (e.g., the encoder and decoder). Then, the interconnect operational frequency is:

$$F(V_{SW}) = \frac{1}{D_{Total}(V_{SW})} = \frac{1}{D_{Error-control} + D_{channel}(V_{SW})} \tag{2}$$

where $D_{Total}(V_{SW})$ is the total delay of the interconnect caused by both the channel and error-control circuit.

Suppose L bits are put into K flits of length L_F bits. Since each flit is transmitted in one cycle, the time required for transmitting a flit is $D_{Total}(V_{SW})$; hence, the maximum number of flits which can be transmitted during the time interval T is:

$$M(V_{SW}) = \left\lfloor \frac{T}{D_{Total}(V_{SW})} \right\rfloor = \lfloor T \cdot F(V_{SW}) \rfloor \tag{3}$$

When a flit is transmitted over an on-chip network interconnect, the following three cases are possible to happen: **Case 1** *(Correct flit):* In this case, the flit is either fault-free or with a fault that can be corrected in the receiver without requiring any retransmission. **Case 2** *(Retransmission requiring flit):* In this case, a fault occurs in the transmitted flit but the error-control scheme detects the fault and initiates a retransmission of the flit. **Case 3** *(Residual faulty flit)*: In this case, a fault occurs in the flit which cannot be tolerated by the error-control scheme. The probability of this happening sometimes is referred to as Residual Error Probability [2], [3]. This happens when either 1) the error-control scheme detects a fault but cannot tolerate it, because for example the scheme does not support retransmissions, or 2) a fault occurs but the error-control scheme cannot detect it, hence no action is taken to tolerate the fault.

Let *c*, *r*, and *f* be the probabilities of *Case 1*, *Case 2*, and *Case 3* respectively. Since all the possibilities have been considered above, we can write: $c+r+f=1$. As shown in the following, the probabilities *c*, *r*, and *f* are used to develop performability models for error-control schemes.

Consider the schemes with retransmission capability (i.e., ARQ and HARQ). Suppose that the transmission of L useful bits (put into K flits) within the time interval T is finished successfully and exactly i faulty flit(s) occur during this transmission. None of these i faulty flits can be a 'Residual faulty flit' (Case 3) and they all should be 'retransmission requiring flits' (Case 2), because it is supposed that the transmission is finished successfully. Since the retransmission policy is considered to be the Go-Back-N policy, the occurrence of these i faulty flits results in $i{\cdot}N$ more flit transmissions. Therefore, in this case $K+i{\cdot}N$ flit transmissions are required. As mentioned in Section II-A, when a faulty flit occurs, the receiver discards the N-1 received flits following the detected faulty flit regardless of whether they were correct or not. In fact, it is not important at all whether these N-1 flits are correct (Case 1), retransmission requiring (Case 2), or residual faulty (Case 3), since they will be discarded anyway and the receiver will never use them. Therefore, in this paper these N-1 flits are called *discarded flits*. Because of the occurrence of exactly i faulty flits, totally $i{\cdot}(N\text{-}1)$ flits are discarded. From the remaining K+i non-discarded flits:

a) None of them can be a 'Residual faulty flit' (Case 3), because if even one 'Residual faulty flit' occurs, the transmission will fail.

b) The last non-discarded flit which is the (K+i)th non-discarded flit should be a correct flit (Case 1). Otherwise, the (K+i)th non-discarded flit is a retransmission requiring flit (Case 2), which means that more flit transmissions are required and hence the (K+i)th non-discarded flit is not the last non-discarded flit. Note that the probability of the (K+i)th non-discarded flit being correct is: $P1=c$.

c) From the remaining (K+i)-1 non-discarded flits, K-1 flits should be correct flits (Case 1) because in total we require that K flits be transmitted successfully. Also the remaining [(K+i)-1]-(K-1)=i flits should be retransmission requiring flits (Case 2), because it is supposed that exactly i faulty flit(s) occur during the transmission. Assuming that all transmitted flits are independent and equally probable to be a correct flit, a retransmission requiring flit, or a residual faulty flit, the probability that K-1 flits out of (K+i)-1 flits are correct flits and the remaining i flits are retransmission requiring flits is:

$$P2=\begin{pmatrix} K+i-1 \\ K-1 \end{pmatrix}\cdot c^{K-1}\cdot r^{i} \tag{4}$$

Therefore, the probability that the transmission (of L useful bits which are put into K flits) is finished successfully while exactly i faulty flit(s) occur during the transmission is:

$$P(i)=P1\cdot P2=c\cdot\begin{pmatrix} K+i-1 \\ K-1 \end{pmatrix}\cdot c^{K-1}\cdot r^{i}=\begin{pmatrix} K+i-1 \\ K-1 \end{pmatrix}\cdot c^{K}\cdot r^{i} \tag{5}$$

Based on Eq. 3, the maximum number of flits which can be transmitted during the time interval T is $M(V_{SW})$, hence K+iN≤$M(V_{SW})$. Therefore, the maximum number of faulty flits that may occur during this transmission is:

$$\max(i)=\left\lfloor \frac{M(V_{SW})-K}{N} \right\rfloor \tag{6}$$

Based on the definition of interconnect performability, the performability $P(L,T)$ of the error-control schemes which have the retransmission capability (HARQ and ARQ) can be expressed as the probability that the transmission of L useful bits (put into K flits) within the time interval T is finished successfully despite the occurrence of i faulty flit(s), where i can change from 0 to max(i). Based on Eqs. 5 and 6, this performability can be written as:

$$P(L,T)=\sum_{i=0}^{\max(i)}P(i)=\sum_{i=0}^{\left\lfloor \frac{M(V_{SW})-K}{N} \right\rfloor}\begin{pmatrix} K+i-1 \\ K-1 \end{pmatrix}c^{K}r^{i} \tag{7}$$

In the schemes which do not have the retransmission capability (FEC and SNFT), when $K{>}N(V_{SW})$, this means that there is not enough time to transmit K flits during the time interval T, and therefore performability is 0. On the other hand, when $K{\leq}N(V_{SW})$, there is enough time to transmit K flits, however each flit can only be transmitted once and there is no retransmission. Therefore, the transmission of the K flits will be successful if and only if the only transmission of each flit is correct (Case 1), whose probability is c^{K}. Therefore, the performability of FEC and SNFT is:

$$P(L,T)=\begin{cases} 0 & K>M(V_{SW}) \\ c^{K} & K\leq M(V_{SW}) \end{cases} \tag{8}$$

As it can be seen from Eqs. 7 and 8, to evaluate the performability of an interconnect we need to know the c, r, and f probabilities. These probabilities in turn depend on the Bit Error Rate (BER) (i.e., the probability that a transmitted bit will be received in error). In the context of on-chip network interconnects, the relevant literature mostly uses Gaussian noise model to evaluate BER [2], [3], [7]. In this model, it is assumed that all the noise sources collectively induce a noise voltage V_N on the channel which follows a Gaussian distribution with zero mean and variance ${\sigma_N}^2$. Therefore, the BER is given by:

$$BER(V_{SW})=Q\left(\frac{V_{SW}}{2\sigma_N}\right) \tag{9}$$

$$Q(x)=\frac{1}{\sqrt{2\pi}}\int_{x}^{\infty}e^{\frac{-u^2}{2}}du \tag{10}$$

where V_{SW} is the voltage swing.

For each scheme (SNFT, ARQ, FEC, and HARQ) we have analyzed the probabilities *c*, *r*, and *f* as follows:

SNFT scheme

In SNFT, a flit will be a correct flit if and only if all of its bits are correct and intact, therefore the probability of a flit being a correct flit is:

$$c_{SNFT}(V_{SW}) = [1 - BER(V_{SW})]^{L_{SNFT}} \tag{11}$$

where L_{SNFT} is the flit size. Since SNFT does not have the retransmission capability, we have $r_{SNFT}(V_{SW})$=0 and hence:

$$f_{SNFT}(V_{SW}) = 1 - [1 - BER(V_{SW})]^{L_{SNFT}} \tag{12}$$

ARQ scheme

Cyclic redundancy check (CRC) codes are error detecting codes that are widely used in communications links [13] and in particular are used for implementing ARQ for on-chip interconnects [2], [3], [9]. Similarly, in this paper we consider the ARQ schemes which are based on CRC codes. In ARQ, like in SNFT, a flit will be a correct flit if and only if all of its bits are correct, therefore:

$$c_{ARQ}(V_{SW}) = [1 - BER(V_{SW})]^{L_{ARQ}} \tag{13}$$

where L_{ARQ} is the flit size in ARQ. It has been shown that the residual error probability of a CRC code can be expressed as [13]:

$$f_{ARQ}(V_{SW}) = A_{d_{\min}} \cdot BER(V_{SW})^{d_{\min}} \tag{14}$$

where $d_{\min}$ is the minimum Hamming distance of the CRC code, and $A_{d_{min}}$ is the number of code words with weight $d_{\min}$. For a CRC code, the $d_{\min}$ and $A_{d_{min}}$ parameters depend on the generator polynomial [13] and the flit size. In this paper, in all experiments and case studies, it is assumed that each flit contains 32 bits, excluding the check bits. Also, in all experiments and case studies (Section III), we consider a CRC code with the generator polynomial $x^8+x^5+x^4+x^3+1$ (called DARC-8 [14]). Therefore, we developed a software code to evaluate the $d_{\min}$ and $A_{d_{min}}$ parameters for this CRC code, and we obtained: $d_{\min}$=2, $A_{d_{min}}$=29. Based on Eqs. 13 and 14, we have:

$$\begin{aligned} r_{ARQ}(V_{SW}) = 1 - [1 - BER(V_{SW})]^{L_{ARQ}} \\ - A_{d_{\min}} \cdot BER(V_{SW})^{d_{\min}} \end{aligned} \tag{15}$$

FEC scheme

For FEC, a flit is considered faulty when it has more than one erroneous bit. Those flits which have only one erroneous bit are not considered as faulty flits, since they are recoverable by the receiver. Therefore the probability of a flit being a correct flit is:

$$\begin{aligned} c_{FEC}(V_{SW}) = [1 - BER(V_{SW})]^{L_{FEC}} + \\ L_{FEC} \cdot BER(V_{SW}) \cdot [1 - BER(V_{SW})]^{L_{FEC}-1} \end{aligned} \tag{16}$$

where L_{FEC} is the flit size in FEC. Since FEC does not have the retransmission capability, we have $r_{FEC}(V_{SW})$=0 and hence:

$$\begin{aligned} f_{FEC}(V_{SW}) = 1 - [1 - BER(V_{SW})]^{L_{FEC}} \\ - L_{FEC} \cdot BER(V_{SW}) \cdot [1 - BER(V_{SW})]^{L_{FEC}-1} \end{aligned} \tag{17}$$

HARQ scheme

For HARQ, like FEC, a flit is considered faulty when it has more than one erroneous bit. Hence, the probability of a flit being a correct flit is:

$$\begin{aligned} c_{HARQ}(V_{SW}) = [1 - BER(V_{SW})]^{L_{HARQ}} \\ + L_{HARQ} \cdot BER(V_{SW}) \cdot [1 - BER(V_{SW})]^{L_{HARQ}-1} \end{aligned} \tag{18}$$

where L_{HARQ} is the flit size in HARQ. Assuming that the error correction code can also be used for double-bit error detection (e.g., overlapping parity bits [2]), the residual error probability can be expressed as [8]:

$$\begin{aligned} f_{HARQ}(V_{SW}) = \\ \sum_{j=1}^{\left\lfloor \frac{L_{HARQ}}{2} \right\rfloor} \binom{L_{HARQ}}{2j+1} BER(V_{SW})^{2j+1} \cdot [1 - BER(V_{SW})]^{L_{HARQ}-2j-1} \end{aligned} \tag{19}$$

and hence:

$$\begin{aligned} r_{HARQ}(V_{SW}) = 1 - [1 - BER(V_{SW})]^{L_{HARQ}} \\ - \sum_{j=0}^{\left\lfloor \frac{L_{HARQ}}{2} \right\rfloor} \binom{L_{HARQ}}{2j+1} BER(V_{SW})^{2j+1} \cdot [1 - BER(V_{SW})]^{L_{HARQ}-2j-1} \end{aligned} \tag{20}$$

TABLE I

POWER, ENERGY, AND DELAY OF ERROR-CONTROL HARDWARE*

Error control circuitry		Static Power$^{\Psi}$ (nW)	Dynamic Power (nW)	Total dynamic energy† (fJ)	Dynamic energy per flit† (fJ/flit)	Dynamic energy per useful bit† (fJ/ubit)	Circuit delay (ns)
CRC (DARC-8)	Encoder	9589	14326	8994.1	140.5	4.3906	0.81
	Decoder	5988	7633	4792.0	74.9	2.3406	1.17
Overlapping Parity (FEC)	Encoder	6023	8952	5620.5	87.8	2.7437	0.78
	Decoder	6463	8981	5638.6	88.1	2.7531	1.64
Overlapping Parity (HARQ)	Encoder	10453	15420	9680.8	151.3	4.7281	0.90
	Decoder	6697	8999	5649.7	88.3	2.7594	1.76

* 2^{11} useful bits were put into 2^6 flits, each containing 32 useful bits

† Dynamic energy per useful bit has been calculated, since it should be inserted in Eq. 27

Ψ Static Power has been estimated, since it should be inserted in Eq. 26

C. Energy consumption model

The dynamic energy consumption of an on-chip wire per bit is [4]:

$$E_{link}(V_{SW}) = \alpha \cdot C_L \cdot V_{DD} \cdot V_{SW} \tag{21}$$

where α is the switching activity, C_L is the wire capacitance, and V_{DD} is the supply voltage.

It has been observed that when a reduced voltage swing is used, the transistors of the receiver level shifter may never be cutoff because of a low input voltage swing [4]; hence a considerable current flows through the receiver level shifter. This current can be calculated as:

$$I_{REC-Static}(V_{SW}) = \begin{cases} 0 & \frac{V_{DD}}{2} - \frac{V_{SW}}{2} \le V_{th} \\ \frac{\beta}{2}\left(\frac{V_{DD}}{2} - \frac{V_{SW}}{2} - V_{th}\right)^2 & \frac{V_{DD}}{2} - \frac{V_{SW}}{2} > V_{th} \end{cases} \tag{22}$$

where β is the transistor beta parameter, V_{DD} is the supply voltage, and V_{th} is the threshold voltage of the transistors. The energy consumption per bit, dissipated by this current is:

$$E_{REC-Static}(V_{SW}) = \frac{V_{DD} \cdot I_{REC-Static}(V_{SW})}{F(V_{SW})} \tag{23}$$

Another important source of energy consumption in on-chip interconnects is the error-control circuit. The energy consumption of the error-control circuit has two components: static and dynamic. Let P_S be the static power of the error-control circuit. Since each flit is transmitted in one cycle, the static energy consumption per flit is $P_S / F(V_{SW})$, where $F(V_{SW})$ is the interconnect operational frequency given by Eq. 2. Hence, the static energy per bit is:

$$E_{CIR-Stat}(V_{SW}) = \frac{P_S}{L_F \cdot F(V_{SW})} \tag{24}$$

where L_F is the flit size. Let $E_{CIR-Dyn}$ be the dynamic energy consumption per bit. The total energy per bit which is consumed by the error-control circuit can be written as:

$$E_{CIR}(V_{SW}) = E_{CIR-Dyn} + E_{CIR-Stat}(V_{SW}) \tag{25}$$

Note that the dynamic energy consumption per bit is frequency independent, because to process a bit of data a certain number of signal transitions are required regardless of the rate at which the circuit processes data. Considering all the sources of energy consumption (Eqs. 21, 23, and 25), the total energy consumption per bit which is consumed by both the channel and error-control circuit is:

$$E_{tot}(V_{SW}) = E_{link}(V_{SW}) + E_{REC-Static}(V_{SW}) + E_{CIR}(V_{SW}) \tag{26}$$

Suppose that the transmission of L useful bits (put into K flits) within the time interval T is finished successfully. When the Go-Back-N policy is used for the schemes with retransmission capability (ARQ and HARQ), if i faulty flit(s) occur during the transmission, $K+i \cdot N$ flit transmissions will be required (Section II-B). Since the probability that i faulty flit(s) occur during the transmission is $P(i)$ (Eq. 5), the expected number of total flit transmissions (including the original flit transmissions as well as the retransmissions) is:

$$N_T = \sum_{i=0}^{\max(i)} P(i) \cdot (K + i \cdot N) \tag{27}$$

where $\max(i)$ is given by Eq. 6. Therefore, for the retransmission-based schemes (ARQ and HARQ), the expected energy consumption required for the successful transmission of K flits during the time interval T is:

$$E_{\text{RT-based}}(V_{SW}) = N_T L_F E_{tot}(V_{SW}) \tag{28}$$

where L_F is the flit size. In the retransmission-free schemes (FEC and SNFT), each flit is transmitted only once. Therefore, the energy consumption required for the successful transmission of K flits during the time interval T is:

$$E_{\text{RT-free}}(V_{SW}) = K \cdot L_F \cdot E_{tot}(V_{SW}) \tag{29}$$

III. EVALUATION OF THE ERROR-CONTROL SCHEMES

In this section we will evaluate the error-control schemes as well as the non-fault-tolerant one for energy consumption and performability. We first estimate the energy overhead of the error-control circuitries, using SPICE simulations. Then we use the analytical models, developed in Section II to analyze and compare different communication schemes.

A. Energy overhead of error-control circuitry

To analyze the energy overhead of the error-control circuits, we synthesized the error-control circuits into 45nm SPICE models. The simulations were carried out using 45nm PTM technology [10] (V_{DD}=0.5V). Note 45nm technology has been used as a way of an example and the models, developed in this work, are generic and can be used for other technologies. A cyclic redundancy code with the generator polynomial $x^8+x^5+x^4+x^3+1$ [14] was used for ARQ, while overlapping parity methods [6] were used for FEC and HARQ. A CRC circuitry can be easily implemented using a Linear Feedback Shift Register (LFSR). However, the LFSR-based implementation is unsuitable for parallel communication interconnects. Therefore, a Parallel Bit Code Generator [7] (PBCG) method was employed for CRC. The aim of the SPICE experiments was to obtain the energy and power values from the simulation to insert them in the analytical models obtained in Section II-C, i.e., Eqs. 24 and 25. For Eq. 24, we needed to evaluate the static power P_S and for Eq. 25, we needed to evaluate the dynamic energy per bit $E_{CIR\text{-}Dyn}$. For the evaluation of $E_{CIR\text{-}Dyn}$, some random data bits were encoded and decoded. Each flit contained 32 useful bits as well as redundant check bits. It was assumed that all data combinations are equally probable to be transmitted (this is a simplified assumption, but the same methodology can be applied to any data pattern). In order to determine the interconnect operational frequency (Eq. 2) we also needed to evaluate the delay of the error-control circuits. The values of energy consumption and circuit delays were obtained using TRANSIENT SPICE analysis. The simulation results are shown in Table I.

Apparently an error correction circuit should be more complex than an error detection circuit, because an error correction circuit not only detects the faults but also corrects them. However, an error detection circuit with high error detection capability may be even more complex than an error correction circuit with relatively lower error detection capability. For example, consider the error detection and error correction circuits that are considered in this paper, i.e., the DARC-8 and overlapping parity circuits respectively. The DARC-8 circuit is only able to detect errors and cannot correct them; however thanks to its complex hardware, it provides a higher error detection capability than the overlapping parity circuit. In fact, DARC-8 is more effective in detecting multiple-bit errors as compared to the overlapping parity method, so that the residual error probability of the overlapping parity method is worse than that of DARC-8. This is why, in Table I, the energy consumption of the DARC-8 circuit is comparable to that of the overlapping parity circuit. It should be noted that there are various CRC circuitries with different generator polynomials that differ in complexity and detection capability. As compared to CRC circuitries with fairly simple generator polynomials (e.g., x^8+1 considered in [2]), DARC-8 (with the generator polynomial $x^8+x^5+x^4+x^3+1$) has more complex hardware and consumes relatively more power but provides a better error detection capability.

Another noticeable issue which can be seen from Table I is that although both HARQ and FEC use the overlapping parity method, the energy consumption of the HARQ error-control circuit is more than that of the FEC error-control circuit. This is because HARQ requires more hardware resources to provide the retransmission capability. For example, HARQ requires buffering resources to store a copy of those flits that are transmitted and their ACKs are still not received (Go-Back-*N* policy). Note that in this paper it is not intended to provide a study of the hardware complexity (area overhead) of the error control schemes. Some information on the hardware complexity (area overhead) of the error-control schemes can be found in [2] and [9].

B. Analysis of performability/energy trade-off

In this analysis, we make the following assumptions: the wire capacitance is C_L=1pF (a few millimeters long wire in 45nm technology [11]). Threshold, supply voltage, and Gaussian noise are V_{th}=0.11V, V_{DD}=0.5V, and σ_N=0.05V respectively. The amount of data that has to be transmitted consists of L=1120 useful bits, which have been split into K=35 flits, each containing 32 useful bits. It is assumed that these data bits need to be transferred during the time interval T=700ns and all the bits are independent and equally probable to be 0 or 1.

Since DARC-8 has been used for ARQ, the flit size in ARQ is L_{ARQ}=(32+8) bits. Also since overlapping parity methods have been used for HARQ and FEC, the flit size in HARQ and FEC is L_{HARQ}=L_{FEC}=(32+7) bits. Assuming that, in ARQ and HARQ, the channel and the 'Retransmission Request' line shown in Fig. 1 operate in parallel and none of them is pipelined (i.e., at any time instant, just one flit is transmitted over the channel and just one ACK/NACK is transmitted over the 'Retransmission Request' line), the window size for the Go-Back-*N* policy is N=2 (for more information on window size refer to [18]).

Using the analytical models developed in Section II (i.e., Eqs. 7, and 28 for ARQ and HARQ and Eqs. 8, and 29 for FEC and SNFT), Fig. 2 shows the performability/energy trade-off for the communication schemes. This figure shows how the energy consumption and the performability of the communication schemes change as V_{SW} changes. Three main observations are made from Fig. 2:

- The maximum achievable performability (at the maximum voltage swing V_{SW}=0.5V) from SNFT is less than $1\text{-}10^{-4}$, while error-control schemes can provide much better performabilities, i.e., significantly greater than $1\text{-}10^{-4}$. Therefore, the usage of error-control schemes is essential in noisy environments to achieve a highly reliable communication. This observation is in line with previous

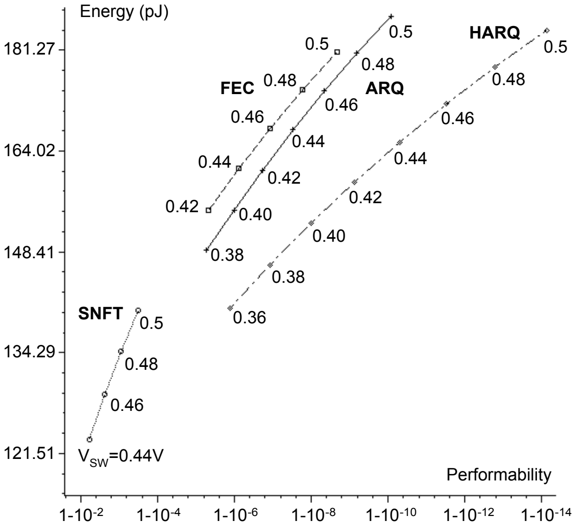


Fig. 2. Performability/energy trade-off

works [2], [3], [9].

- For a given performability constraint, HARQ consumes less energy than ARQ and FEC. For example, if we require a performability of 1-10^{-8}, we can use ARQ with V_{SW}=0.45V. However, if we use HARQ with V_{SW}=0.40V, we will achieve the required performability but with 10.6% energy saving. Note that none of the previous works [2], [3], [9] has reached to the same conclusion.
- While the maximum achievable performability from FEC and ARQ are about 1-10^{-9} and 1-10^{-10} respectively, the maximum achievable performability from HARQ is much higher – about 1-10^{-14}. Again note that none of the previous works [2], [3], [9] has reached to the same conclusion.

Influence of noise power

It has been observed that noise power varies for different applications and environments [3], [19], so that the related literature often considers different ranges of possible noise power values. For example, in [19] two different noise power values, σ_N=0.3V and σ_N=0.5V, are considered for logic gates with V_{DD}=1.5V. As another example, in [3] it is considered that for an on-chip interconnect in a 90-nm technology (with V_{DD}=1V), the noise power varies from 0.04V to 0.1V. In this paper, the intention is not to consider any specific noise power value; rather we aim to analyze how the effectiveness of the error-control schemes change as the noise power changes. Therefore, we consider a wide range of noise power values between two extreme cases. Fig. 3 shows the performability/energy trade-off of the communication schemes when the noise power varies between the following excessively low and excessively high noise power values:

1- σ_N=0.01V (Fig. 3a): In this case the noise is so weak that no error control is required. This is because as it can be seen from Fig. 3a, SNFT can provide a performability of 1-10^{-134}, which is very close to 1. Considering the definition of performability (Section II-B), a performability of 1-10^{-134} means that the transmission of the given amount of data within the given time interval will be finished successfully with the probability of 1-10^{-134}. Since this probability is very close to 1, it is not necessary to improve the performability and hence the use of error-control schemes is unnecessary.

2- σ_N=0.135V (Fig. 3f): In this case the noise is so strong that the interconnect fails despite the use of error-control schemes. For example, it can be seen from Fig. 3f that when σ_N=0.135V, the maximum achievable performability is about 1-$10^{-0.0025}$ = 0.00574 (HARQ, V_{SW} =0.5V). A performability of 0.00574 means that the transmission of the given amount of data within the given time interval will be finished successfully with the probability of 0.00574. This probability is very low and indicates that the interconnect most likely (with a probability of 0.99426) fails.

Two interesting observations can be made from Fig. 3:

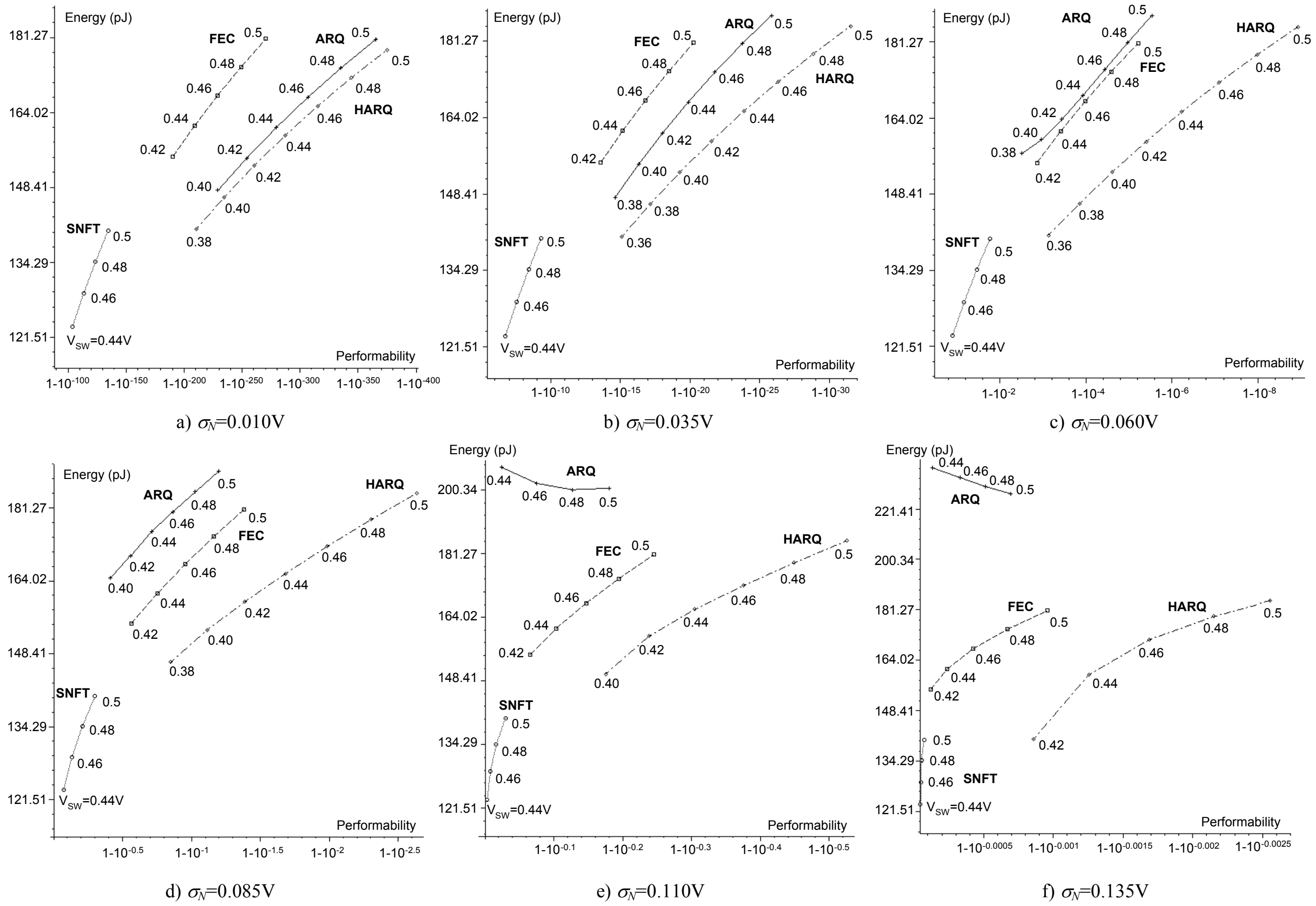
- When the noise power is low (Figs. 3a and 3b), ARQ is more effective than FEC. However as the channel becomes more noisy (Figs. 3c, 3d, 3e, and 3f), ARQ becomes less advantageous than FEC. We clarify this by means of the following example:

  - When σ_N =0.035V (Fig. 3b), if we use FEC with V_{SW}=0.44V, we will achieve a performability of 1-10^{-15}. However, if we use ARQ with V_{SW}=0.40V, we will achieve the same performability but with 4.3% energy saving.
  - When σ_N =0.06V (Fig. 3c), if we use FEC with V_{SW}=0.44V, we will achieve a performability of about 1-10^{-3}. If we use ARQ with V_{SW}=0.42V, we will achieve the same performability but with 1.6% more energy consumption.
  - When σ_N =0.085V (Fig. 3d), if we use FEC with V_{SW}=0.44V, we will achieve a performability of about 1-$10^{-0.8}$. If we use ARQ with V_{SW}=0.44V, we will achieve the same performability but with 9.4% more energy consumption.

  In short, as σ_N increases, the energy saving of FEC over ARQ improves. This is because a strong noise can repeatedly affect the retransmitted flits. Therefore a simple retransmission scheme (i.e., ARQ) is not suitable for a very noisy channel.

- While the maximum achievable performabilities (at V_{SW}=0.5V) decrease with the increase in nose power, the maximum achievable performability from HARQ is always significantly higher than what is achievable from the other schemes. For example, when σ_N =0.06V (Fig. 3c), the maximum achievable performabilities from SNFT, FEC and ARQ are about 1-10^{-2}, 1-10^{-5} and 1-10^{-6} respectively, but the maximum achievable performability from HARQ is about 1-10^{-9}. This shows the importance of HARQ.

Influence of wire length



a) σ_N=0.010V b) σ_N=0.035V c) σ_N=0.060V

d) σ_N=0.085V e) σ_N=0.110V f) σ_N=0.135V

Fig. 3. Influence of noise power

Since the length of interconnects varies for different on-chip networks, a wide range of interconnect capacitances is considered in the related literature. For example, in [2] two different interconnect capacitance values are considered for a 180-nm technology: C_L=0.5pF (a few millimeter long wires in a 180-nm technology) and C_L=5pF (a wire of about 1cm in a 180-nm technology). In [3], a capacitance of 2.73pF is considered for an on-chip interconnect in a 90-nm technology (a wire of about 1cm in a 90-nm technology). In this paper, we do not consider any specific capacitance value; rather we analyze how the effectiveness of the error-control schemes change as the interconnect capacitance (length) changes. For this purpose, we assume that the interconnect capacitance C_L varies from 0.01pF to 1pF. Based on the information provided in [11], in a 45-nm technology, a capacitance of 0.01pF corresponds to an interconnect length of about 0.05mm and a capacitance of 1pF corresponds to an interconnect length of about 5mm. Fig. 4 shows the performability/energy trade-off of the communication schemes when the interconnect capacitance C_L varies from 0.01pF to 1pF. Two main observations are made from Fig. 4:

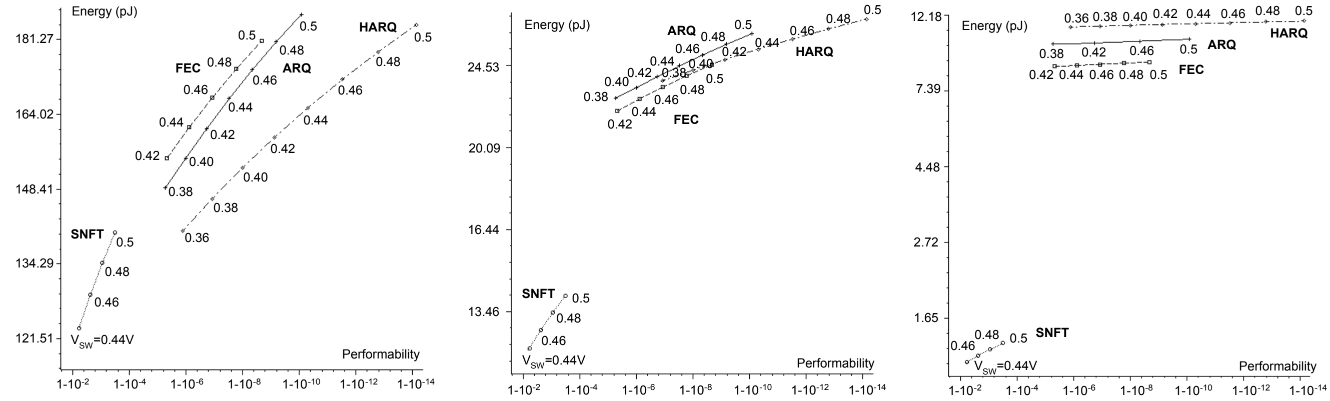
- When C_L =1pF (Fig. 4a), HARQ consumes less energy than ARQ and FEC. However, as the wire capacitance C_L (wire length) decreases (Fig. 4b and 4c), the energy saving of HARQ over ARQ and FEC decreases. We clarify this by means of the following example: Suppose we require a performability of $1\text{-}10^{-8}$. To achieve this level of performability:

- When C_L =1pF (Fig. 4a), we can use ARQ with V_{SW}=0.45V and HARQ with V_{SW}=0.40V. However, at these voltage settings, HARQ offers 10.6% energy saving as compared to ARQ.

- When C_L =0.1pF (Fig. 4b), we can use ARQ with V_{SW}=0.45V and HARQ with V_{SW}=0.40V. However, at these voltage settings, HARQ offers 2.4% energy saving as compared to ARQ. In fact, it can be seen from Fig. 4b that when C_L =0.1pF, the FEC, ARQ and HARQ curves become very close to each other which means that there is no considerable difference between the energy consumption of the three schemes.

- When C_L =0.01pF (Fig. 4c), we can use ARQ with V_{SW}=0.45V and HARQ with V_{SW}=0.40V. In this case, HARQ consumes 11.4% more energy than ARQ.

In short, with the performability constraint of $1\text{-}10^{-8}$, as C_L decreases from 1pF to 0.01pF, the energy saving of HARQ over ARQ decreases from +10.6% to -11.4%. This is mainly because, as it can be seen from Table I, the energy consumption of the HARQ error-control circuit is more than



a) C_L=1pF

b) C_L=0.1pF

c) C_L=0.01pF

Fig. 4. Influence of wire length

that of the ARQ error-control circuit. In the interconnects made up of long wires, the main portion of the energy is consumed by the wires and not by the error-control circuit; hence, the difference between the energy consumption of the ARQ and HARQ error-control circuits is negligible. However, as the wire length decreases, the energy consumption of the error-control circuits becomes a significant portion of the total energy; hence the energy saving of HARQ over ARQ decreases because of the higher energy consumption of the HARQ error-control circuit.

- As the wire capacitance C_L decreases, the slope of the curves decreases so that in Fig. 4c, the curves are close to being horizontal. This means that as C_L decreases, the effectiveness of reducing V_{SW} decreases. For example, in Fig. 4c, when V_{SW} of HARQ decreases from 0.5V to 0.36V, the energy consumption only decreases from 11.73 pJ to 11.29 pJ, while the performability decreases considerably from $1\text{-}10^{-14}$ to $1\text{-}10^{-6}$. This is because, when an interconnect is made up of short wires, the energy consumed by the wires is only a small portion of the total interconnect energy and the main portion of the energy is consumed by the error control circuit. In this case, reducing the voltage swing can only achieve a negligible energy saving, while it still has a considerable negative impact on the interconnect performability.

Influence of time constraints

So far, we have analyzed the performability $P(L$=35*32,T=700ns$)$. Assuming that L is constant, for the applications which do not have tight time constraints, we can analyze the performability for relatively large T values. However, for the applications with tight time constraints, smaller T values have to be considered. In order to study the impact of the time constraints on the efficiency of the error-control schemes, Fig. 5 shows the performability/energy trade-off of the communication schemes when T=355ns, i.e., in Fig. 5, we consider the performability $P(L$=35*32, T=355ns$)$. Two key observations are made from Fig. 5:

- When we compare Fig. 2 (T=700ns) with Fig. 5 (T=355ns), it can be seen that when T=700ns (relaxed time constraint), ARQ is more effective than FEC. However, when T=355ns (tight time constraint), ARQ becomes less advantageous than FEC. For example, when T=355ns, the maximum achievable performability from ARQ is about $1\text{-}10^{-7}$. However, if we use FEC with V_{SW}=0.48V, we will achieve not only a performability more than $1\text{-}10^{-7}$ but also 7% energy saving. This is because ARQ only relies on retransmissions to tolerate faults. Therefore, when tight time constraints are imposed, ARQ has relatively less time to retransmit faulty flits and hence its performability decreases. However, imposing tight time constraints does not have a similar negative impact on FEC, as it does not use retransmissions. [2] has studied energy/reliability trade-off and reported that for the same constraint on system reliability, ARQ consumes less energy than FEC. This is true and our observation is in agreement with it (Fig. 2) but only when we do not require high performance (relaxed time constraints). It can be seen from Fig. 5 that when we require high performance (tight time constraints), ARQ is less effective than FEC.
- When we compare Fig. 2 (T=700ns) with Fig. 5 (T=355ns), it can be seen that when T=700ns (relaxed time constraint), HARQ is more effective than FEC. However, when T=355ns (tight time constraint), HARQ becomes less effective than FEC. In fact, when T=355ns (tight time constraint), HARQ does not have enough time to retransmit faulty flits and hence, just like FEC, it can only correct single-bit errors at the receiver without any retransmissions. Therefore, as it can be seen from Fig. 5, when the voltage swings of FEC and HARQ are the same, they provide almost the same performabilities. Since the energy consumption of the HARQ error-control circuit is more than that of the FEC error-control circuit (Table I), when the voltage swings of both the schemes are the same, although they provide almost the same performabilities, HARQ consumes more energy than FEC.

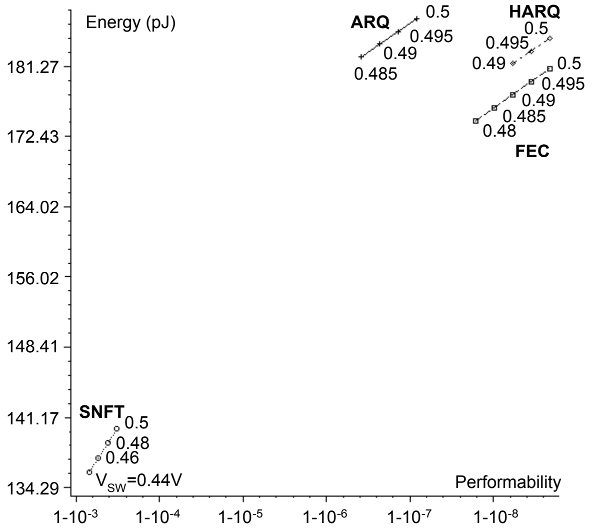


Fig. 5. Performability/energy trade-off for an application with tight time constraints

IV. Concluding Remarks and Future Works

In this paper, we have argued that the use of error-control schemes in on-chip networks results in *degradable systems*, hence performance and reliability must be measured jointly using the 'Performability' metric. We have analyzed the impact of three error-control schemes on the trade-off between performability and energy in on-chip networks, when voltage swing, noise power, wire length (wire capacitance) and time constraint vary. This is unlike the previous works [2], [3], [9] which none of them has addressed the degradable nature of on-chip interconnects and the performability metric.

Since noise power and time constraint vary for different applications and environments, and wire length varies for different on-chip interconnects, the impacts of these three factors (noise power, time constraint, and wire length) on the effectiveness of the error-control schemes have been analyzed in this paper. This analysis shows that:

- The maximum achievable performability (at the maximum voltage swing) from HARQ is always higher than (or almost equal to) what is achievable from the other schemes.

- For a given performability constraint, HARQ consumes less energy than ARQ and FEC, except for when short wires are used, or when tight time constraints are imposed.

- When short wires are used, HARQ provides the best performability and consumes the most energy. Also, FEC provides the least performability and consumes the least energy among the error-control schemes. It is worth mentioning that when short wires are used, reducing the voltage swing is not suitable.

- When tight time constraints are imposed, HARQ and FEC provide almost the same performabilities and can provide better performabilities than ARQ. However, since FEC consumes less energy than HARQ, FEC is preferable to HARQ.

Although we have analyzed a number of factors that have significant impacts on the performability/energy trade-off in the communication schemes (i.e., voltage swing, noise power, wire length, and time constraint), it is clear that there may be other factors that can affect this trade-off. Future work mainly involves analyzing the other factors that may have noteworthy impacts on the performability/energy trade-off in the communication schemes. For instance, it is becoming common in deep submicron designs to use repeaters for on-chip interconnects [17]. These repeaters have an influence on the delay and energy consumption of on-chip interconnects [17]. Therefore, an interesting topic for future work is to investigate the impact of the use of repeaters on the performability/energy trade-off. Another interesting topic for future work is to consider the use of error-control schemes for current-mode interconnects [20] and to analyze their performability/energy trade-offs.

References

[1] L. Benini, and G. De Micheli, "Networks on Chips: A New SoC Paradigm," *Computer*, 35(1), pp. 70-78, 2002.

[2] D. Bertozzi, L. Benini, and G. De Micheli, "Error Control Schemes for On-Chip Communication Links: The Energy–Reliability Tradeoff," *IEEE Trans. CAD*, 24(6), pp. 818-831, 2005.

[3] F. Worm, P. Ienne, P. Thiran, and G. De Micheli, "A Robust Self-Calibrating Transmission Scheme for On-Chip Networks," *IEEE Trans. VLSI*, 13(1), pp. 126-139, 2005.

[4] C. Svensson, "Optimum Voltage Swing on On-Chip and Off-Chip Interconnect," *IEEE JSSC*, 36(7), pp. 1108-1112, 2001.

[5] R. M. Smith, K. S. Trivedi, and A. V. Ramesh, "Performability Analysis: Measures, an Algorithm, - and a Case Study," *IEEE Trans. Computers*, 37(4), pp. 406-417, 1988.

[6] D. K. Pradhan, *Fault-Tolerant Computing: Theory and Techniques,* Prentice Hall, 1986.

[7] D. Bertozzi, L. Benini, and G. De Micheli, "Error Control Schemes for On-chip Interconnection Networks: Reliability versus Energy Efficiency," *Networks on Chip*. A. Jantsch, H. Tenhunen, Eds. Kluwer, 2003.

[8] D. Bertozzi, L. Benini, and G. De Micheli, "Low Power Error Resilient Encoding for On-Chip Data Buses," *Proc. DATE 2002*, pp. 102-109, 2002.

[9] S. Murali, T. Theocharides, N. Vijaykrishnan, M. J. Irwin, L Benini, and G. De Micheli, "Analysis of Error Recovery Schemes for Networks on Chips," *IEEE Design & Test of Computers*, 22(5), pp. 434-442, 2005.

[10] W. Zhao, and Y. Cao, "New Generation of Predictive Technology Model for Sub-45nm Design Exploration," *Proc. ISQED 2006*, pp. 585-590, 2006.

[11] C. Grecu, P. P. Pande, A. Ivanov, and R. Saleh, "A Scalable Communication-Centric SoC Interconnect Architecture," *Proc. ISQED 2004*, pp. 343-348, 2004.

[12] J. F. Meyer, "On Evaluating the Performability of Degradable Computing Systems," *IEEE Trans. Computers*, c-29(8), pp. 720-731, 1980.

[13] G. Castagnoli, J. Ganz, and P. Graber, "Optimum Cyclic Redundancy-Check Codes with 16-Bit Redundancy," *IEEE Trans. Communications*, 38(1), pp. 111-114, 1990.

[14] P. Koopman, and T. Chakravarty, "Cyclic Redundancy Code (CRC) Polynomial Selection for Embedded Networks," *Proc. Int'l Conf. DSN 2004*, pp. 145-154, 2004.

[15] D. Park, C. Nicopoulos, J. Kim, N. Vijaykrishnan, and C. R. Das, "Exploring Fault-Tolerant Network-on-Chip Architectures," *Proc. Int'l Conf. DSN 2006*, pp. 93-104, 2006.

[16] C. Grecu, A. Ivanov, R. Saleh, E. S. Sogomonyan, and P. P. Pande, "On-Line Fault Detection and Location for NoC Interconnects," *Proc. Int'l Symp. IOLTS 2006*, pp. 145-150, 2006.

[17] L. Zhang, H. Chen, B. Yao, K. Hamilton, and C. K. Cheng, "Repeated On-Chip Interconnect Analysis and Evaluation of Delay, Power, and Bandwidth Metrics under Different Design Goals," *Proc. Int'l Symp. ISQED 2007*, pp. 251-256, 2007.

[18] A. S. Tanenbaum, *Computer Networks*, 4th Edition, Prentice-Hall, 2003.

[19] R. Hegde, and N. R. Shanbhag, "Toward Achieving Energy Efficiency in Presence of Deep Submicron Noise," *IEEE Trans. VLSI*, 8(4), pp. 379-391, 2000.

[20] A. P. Jose, G. Patounakis, and K. L. Shepard, "Pulsed Current-Mode Signaling for Nearly Speed-of-Light Intrachip Communication," *IEEE JSSC*, 41(4), 2006.

[21] J. Kim, C. Nicopoulos, D. Park, V. Narayanan, M. S. Yousif, and C. R. Das, "A Gracefully Degrading and Energy-Efficient Modular Router Architecture for On-Chip Networks," *Proc. Int'l Symp. ISCA 2006*, pp. 4-15, 2006.

Alireza Ejlali is an Assistant Professor of Computer Engineering at Sharif University of Technology, Tehran, Iran. He received a Ph.D. degree in computer engineering from Sharif University of Technology in 2006. From 2005 to 2006, he was a visiting researcher in the Electronic Systems Design Group, University of Southampton, UK.

His research interests include low power design, dependability evaluation, and fault tolerant embedded systems.

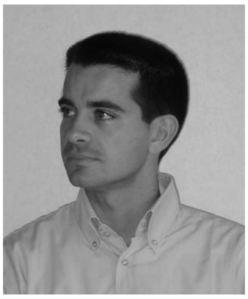
Bashir M. Al-Hashimi (M'99-SM'01) received the B.Sc. degree (with 1st-class classification) in Electrical and Electronics Engineering from the University of Bath, UK, in 1984 and the Ph.D. degree from York University, UK, in 1989. Following this he worked in the microelectronics design industry and in 1999, he joined the School of Electronics and Computer Science, Southampton University, UK, where he is currently a Professor of Computer Engineering and Director of the Pervasive System Center. He has authored one book on SPICE simulation, (CRC Press, 1995), and coauthored two books, Power Constrained Testing of VLSI circuits (Springer, 2002), and System-Level Design Techniques for Energy-Efficient Embedded Systems (Springer, 2004). In 2006, he edited the book, System-on-Chip: Next Generation Electronics (IEE Press, 2006). He has published over 200 papers in journals and refereed conference proceedings. His current research interests include low-power system-level design, system-on-chip test, and reliable nano design.

Prof. Al-Hashimi is a Fellow of the IEE and a Senior Member of the IEEE. He is the Editor-in-Chief of the IEE Proceedings: Computers and Digital Techniques, an editor of the Journal of Electronic Testing: Theory and Applications (JETTA), and is a member of the editorial board of the Journal of Low Power Electronics, and the Journal of Embedded Computing. He was the General Chair of the 11th IEEE European Test Symposium (UK 2006) and he is the Technical-Programme Chair of DATE 09. He is the coauthor of the James Beausang Best Paper Award at the 2000 IEEE International Test Conference relating to low power BIST for RTL data paths, and a co-author of a paper on test data compression which has recently been selected for a Springer book featuring the most influential work over the 10 years of the Design Automation and Test in Europe (DATE) conference.

Paul Rosinger received the B.Sc. in Computer Science from the Technical University of Timisoara, Romania, in 1999, and the Ph.D. in Electronics and Computer Science from the Southampton University, United Kingdom, in 2003. He had been on a postdoctoral research fellowship at Southampton University until December 2007. His research interests included testing of digital systems, low power design and hardware and software reliability. He is currently a researcher at Winton Capital Management.

Seyed Ghassem Miremadi (M'93) is an Associate Professor of Computer Engineering at Sharif University of Technology. As fault-tolerant computing is his specialty, he initiated the "Dependable Systems Laboratory" at Sharif University in 1996 and has chaired the Laboratory since then. The research laboratory has participated in several research projects which have led to several scientific articles, conference papers and technical reports. Dr. Miremadi and his group have done research in Physical, Simulation-Based and Software-Implemented Fault Injection, Dependability Evaluation Using HDL Models, Fault-Tolerant Embedded Systems and Fault Tree Analysis.

Dr. Miremadi was the Education Director (1997-1998), the Head (1998-2002), and the Research Director (2002-2006) of Computer Engineering Department at Sharif University. He is currently the Director of the IT program at International Campus of Sharif University. He got his MSc in Applied Physics and Electrical Engineering from Linköping Institute of Technology and his Ph.D. in Computer Engineering from Chalmers University of Technology, Sweden, in 1984 and 1995, respectively. He served as the general chair of the 13th Int'l CSI Computer Conference (CSICC 2008). He is a member of the IEEE Computer Society, IEEE Reliability Society and the Computer Society of Iran.



Luca Benini (S'94–M'97–SM'04–F'07) is a Full Professor at the Department of Electrical Engineering and Computer Science (DEIS) of the University of Bologna. He also holds a visiting faculty position at the Ecole Polytechnique Federale de Lausanne. He received a Ph.D. degree in electrical engineering from Stanford University in 1997.

Dr. Benini's research interests are in the design of system-on-chip platforms for embedded applications. He is also active in the area of energy-efficient smart sensors and sensor networks, including biosensors and related data mining challenges. He has published more than 350 papers in peer-reviewed international journals and conferences, four books and several book chapters. He has been general chair and program chair of the Design Automation and Test in Europe Conference. He has been a member of the technical program committee and organizing committee of several conferences, including the Design Automation Conference, International Symposium on Low Power Design, and the Symposium on Hardware-Software Codesign.

He is Associate Editor of several international journals, including the IEEE Transactions on Computer Aided Design of Circuits and Systems, the ACM Journal on Emerging Technologies in Computing Systems, and the ACM Transactions on Embedded Computing Systems. He is a Fellow of the IEEE.