

UC Irvine

UC Irvine Previously Published Works

Title

Equi-Noise: A Statistical Model That Combines Embedded Memory Failures and Channel Noise

Permalink

<https://escholarship.org/uc/item/3wp0j0cm>

Journal

IEEE Transactions on Circuits and Systems I Regular Papers, 61(2)

ISSN

1057-7130

Authors

Khairy, Muhammad S
Khajeh, Amin
Eltawil, Ahmed M
et al.

Publication Date

2014-02-01

DOI

10.1109/tcsi.2013.2268197

Peer reviewed

Equi-Noise: A Statistical Model That Combines Embedded Memory Failures and Channel Noise

Muhammad S. Khairy, *Student Member, IEEE*, Amin Khajeh, *Member, IEEE*, Ahmed M. Eltawil, *Member, IEEE*, and Fadi J. Kurdahi, *Fellow, IEEE*

Abstract—This paper exploits the predominance of embedded memories in current and emerging wireless transceivers as a means to save power via channel state aware voltage scaling. The paper presents a statistical model that captures errors in embedded memories due to voltage over-scaling and maps the errors to a Gaussian distribution that represents a combination of communication channel noise and hardware noise. Designers can use the proposed model to investigate different power management policies, that capture the performance of the system as a function of both channel and hardware dynamics, thus creating a much richer design space of power, performance and reliability. A case study of a DVB receiver is presented and the validity of the proposed model is confirmed by simulations.

Index Terms—Embedded memories, fault tolerant, low power, SRAM, wireless communications.

I. INTRODUCTION

DESIGNERS of next generation Systems-on-Chip (SoCs) face daunting challenges in generating high yielding architectures that integrate vast amounts of logic and memories in a minimum die size, while simultaneously minimizing power consumption. Traditional design approaches attempt to guarantee 100% error-free SoCs using a number of fault-tolerant architectural and circuit techniques. However, advanced manufacturing technologies will render it economically impractical to insist on 100% error-free SoCs in terms of area and power [1]–[4].

Fortunately, many important application domains (e.g., communication and multimedia) are inherently error-aware, allowing a range of designs with a specified Quality of Service (QoS) to be generated for varying amounts of error in the system. However, exploitation of error-aware design to address these power, yield and cost challenges requires a significant shift from error-free to error-aware design methodologies [5]–[10].

Manuscript received January 12, 2013; revised April 10, 2013; accepted April 28, 2013. Date of publication July 09, 2013; date of current version January 24, 2014. This work was supported in part by the National Science Foundation under Grant ECCS-0955157. This paper was recommended by Associate Editor J. Ma.

M. S. Khairy, A. M. Eltawil, and F. J. Kurdahi are with the Electrical Engineering and Computer Science Department at the University of California Irvine, Irvine, CA 92697 USA (e-mail: {mkhairy, aeltawil, kurdahi}@uci.edu).

A. Khajeh is with Intel Labs, Hillsboro, OR 97124 USA (e-mail: amin.khajeh@intel.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSI.2013.2268197

In communication and multimedia systems, embedded memories are perfect candidates for this exploration, since the share of the SoC that is dedicated to memories has experienced an increasingly upwards trend exceeding more than 50% of the area of an SoC for wireless standards such as DVB, LTE and WiMAX [11]–[16]. Furthermore, a large portion of the memory is typically used for buffering data that already has a high level of redundancy (e.g. buffering memories in wireless chips, decoded picture buffer in H.264 etc.). Finally, from a network perspective, buffering memories are transparent across a hierarchy since they do not change the nature of the data stored, which allows for simple and efficient cross-layer techniques.

With their high density and aggressive design margins, memories typically suffer from a limited range of voltage scaling, which, in turn reduces the power savings and results in increased design margins to ensure a high level of reliability. However, in the context of wireless communication systems, given that the incoming buffered data is already corrupted by time varying noise and interference, there is no need to store the data samples in memories that are error free, 100% of the time. Rather, Voltage Over Scaling (VOS) techniques ([5], [9], [17]) that are channel state aware, can be used to tradeoff reliability versus power savings as a function of the time varying quality of the incoming data (channel state), as long as the signal to noise ratio at the non-linear decision device is maintained at a desirable level. In prior work, the authors have shown that utilizing fault tolerant techniques on embedded memories (mainly through aggressive voltage scaling) will result in a) 20%–35% power reduction in wireless systems depending on the application, b) savings in cost and area by reducing or eliminating the need for circuit redundancy, and c) achieving a higher “effective yield” by tolerating errors at the system level while keeping other parameters constant [10], [18].

While the gains are lucrative, accurately evaluating the impact of hardware errors on system performance is a challenge. Typically, hardware error statistics for a certain operation conditions (supply voltage, frequency) are gathered and used in a system simulation to evaluate the effectiveness and to quantify the gains of the proposed fault tolerant technique in terms of power savings and system performance impact. This approach suffers from the following major drawbacks:

- i- *Lack of scalability*: Clearly the design space is very large given the numerous possible combinations of system settings and operation conditions. Since each simulation result is valid only for a specific simulation setup, therefore, for every change in the algorithm or policy, a new system

simulation has to be performed, which limits the design space.

- ii- *Accuracy and simulation time*: The accuracy of the obtained results depends on the size of the processed data.

It is therefore necessary to devise accurate analytical models that abstract the underlying fault mechanisms of embedded memories accurately and rapidly. Towards that end, in this paper we propose an efficient power policy based on a joint statistical modeling for both channel and hardware dynamics which we will refer to as the “Equi-Noise technique”. The proposed model in this paper enables engineers and system designers to apply different power management techniques on embedded memories and easily trade-off the degradation in the system performance with the obtained gain in power savings. We illustrate the proposed model in the paper by considering a typical OFDM-based communication system. However, the same concept can be applied to any other wireless communication system. A preliminary version of this work appeared in [19], [20], in which we introduced the concept of modeling the embedded memory noise for communication system. In [19] we presented a statistical model that captures bit failures in embedded memories and quantified the impact of propagating the noise distribution through a finite impulse response filter, while in [20] we investigated the impact of FFT on the noise distribution. In this paper we assume that the data is stored in memories in the form of two’s complement. In [21], a new representation of data for unreliable memories was presented where bit-cells of unreliable buffers are modeled as “stuck-at channel”. While enhancing performance, this technique requires sophisticated algorithms to identify the best data mapping for data words of 4 bits or more and requires new mapping and demapping stages at the input and output of the buffering memories.

The key contribution of this paper is to address the challenge of accurately and rapidly estimating the change in the statistical distribution of data at each block in the communication system leading to or originating from a memory that is experiencing voltage scaling induced errors. By replacing the traditional noise model in communication systems with the developed “Equi-Noise” model, one can investigate different power management policies, where the faulty hardware can be treated as error-free hardware.

The remainder of this paper is organized as follows: Section II presents the problem formulation. Section III presents a comprehensive study of the effect of buffering memory failures, FIR filtering, FFT and channel equalization on the communication channel noise distribution. Section IV presents “Equi-Noise”, a framework that links the failure statistics of embedded memories with the statistical nature of the wireless communication channel. Memory sensitivity analysis is presented in Section V. A power management policy based on the “Equi-Noise” model is discussed in Section VI. A case study of DVB system is presented in Section VII and finally, conclusions are drawn in Section VIII.

II. PROBLEM FORMULATION

While the wireless channel is a stochastic channel where the designer has little control on its variables, one can consider em-

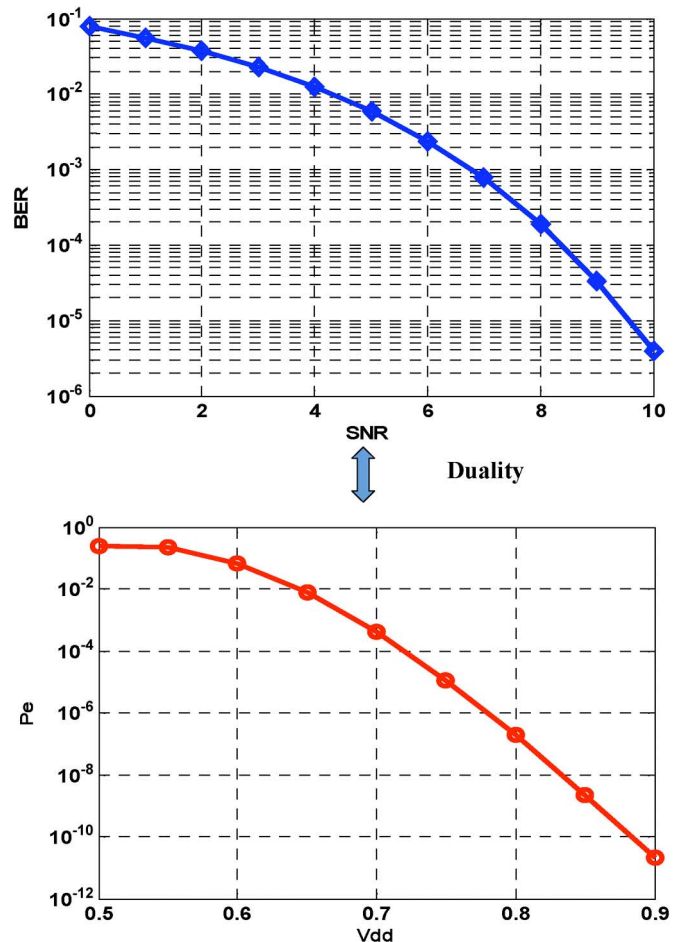


Fig. 1. Duality between a wireless channel and an embedded memory.

bedded memories as an extension of the channel where the designer can control the quality via voltage scaling. Fig. 1 demonstrates the duality between a wireless channel and an embedded memory. In the case of the wireless channel, the Signal to Noise Ratio (SNR) dictates the Bit Error Rate (BER) performance. While, in the case of embedded memories, (experimental measurements for a 65 nm CMOS technology memory with nominal supply voltage of 0.9 v), the supply voltage dictates the error rate of the retrieved data. Traditional techniques attempt to guarantee virtually no error from the hardware channel by assigning a large amount of design margin, or in other words by “over-designing”. The target of this work is to provide a rapid means of identifying and propagating the impact of embedded memory failures through the communication system, thus allowing the designer to opportunistically increase the noise contribution of the hardware channel based on the observed statistics of the actual communication channel to meet a certain metric of quality, such as target BER for communication devices.

To better illustrate the concept of hardware channel noise, we set up a simulation for a simple communication system as shown in Fig. 2. We assumed Additive White Gaussian Noise (AWGN) for the communication channel. We used the models presented in [22]–[24], for the memory errors model. The memory error locations are spatially uniformly distributed and exponentially increase with supply voltage reduction. For simplicity, we assumed binary phase shift keying (BPSK) modulation. Fig. 3 de-

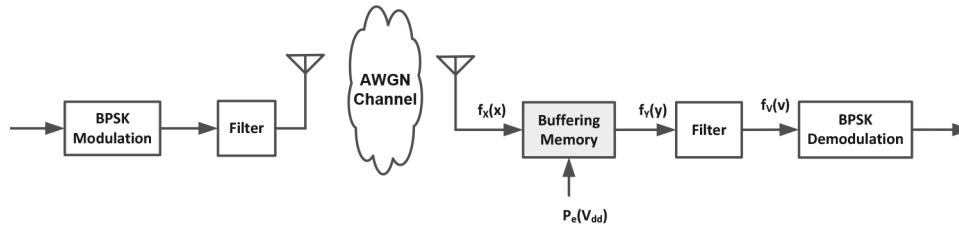


Fig. 2. A simplified wireless communication system model.

 TABLE I
 NOTATIONS AND PARAMETERS DESCRIPTION

Notation	Description	Notation	Description
V_1, V_2	Supply voltage for memory 1 and memory 2	P_{e1}, P_{e2}	Introduced error rate in memory 1 and memory 2
N	Number of bits per memory word	\mathcal{N}	Normal/Gaussian distribution
N_{FFT}	FFT size	$f_X(x)$	Received data distribution
$f_Y(y)$	Data distribution after first buffering memory	$f_V(v)$	Data distribution after filtering
$\varphi_V(i\omega)$	Characteristics function of filtered data	$f_Z(z)$	Data distribution after FFT
$f_{R S}(r s)$	Data distribution after equalization	$f_{U S}(u s)$	Data distribution after second buffering memory
$\bar{\gamma}$	Average channel power	σ_{eq}^2	Equivalent channel noise variance

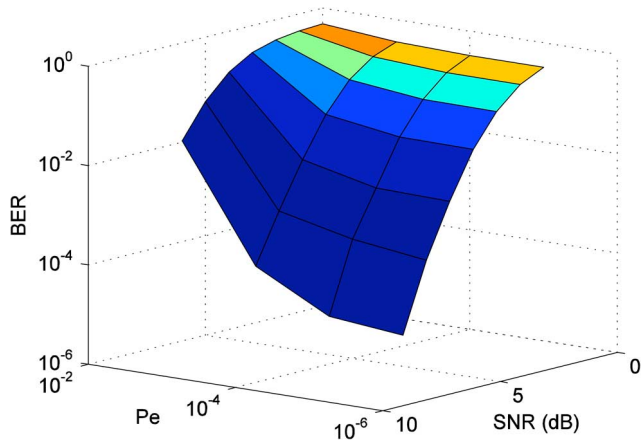


Fig. 3. Hardware can be considered as a voltage controlled channel.

picts the results which indicate that for a given BER, there exist sets of (SNR, P_e) which satisfy this BER. Thus, the memory error rate (P_e) can be adjusted based on the received SNR to meet a given quality metric. Alternatively, one can estimate the effect of the memory error on the bit error rate as an extra contribution to the channel noise when compared to an identical system with error free memory. The estimated “*hardware noise*” is a function of the buffering memory supply voltage (or buffering memory errors, P_e s) and the location of the memory within the system. To generate this abstract model, it is important to first quantify how different key blocks in the communication system chain shape the data and noise distribution.

III. EFFECT OF MEMORY ERRORS ON DIFFERENT SYSTEM BLOCKS

Generally speaking, the performance of SRAM circuits under supply scaling and process variation is well understood. For example, the analysis presented in [24] and the references therein confirm that the access time follows a Gaussian distribution that can be related to the applied supply voltage and the underlying variations in threshold voltage. This section begins by

discussing a mathematical model for memory errors, followed by the propagation of the distribution through communication building blocks such as filters and FFT units, a zero forcing receiver (as an example), culminating with an entire receiver. We consider a generic OFDM-based wireless system with two buffering memories as shown in Fig. 4. Table I presents a summary of the parameters and the notations used in this system. It is worth mentioning that VoS technique is applied to buffering memories that store data payload of the packet while memories within synchronization loops are protected to guarantee correct operation of the receiver.

A. Effect of Memory Errors on Buffering Memories

Fig. 5 shows a simplified version of a typical data buffering memory in a communication system where the red squares represent erroneous memory cells due to voltage overscaling. The wireless communication channel introduces additive white Gaussian noise (AWGN) and the buffering memory introduces random uniform bit-flips depending on the supply voltage. The goal is to find $f_Y(y)$ and Y_n based on the information from the AWGN channel (namely noise power) and $P_e(V_{dd})$ given the distribution of $f_X(x)$. In general, data stored as binary numbers in memory can contain both decimal and fractional parts; each data element stored is denoted by a random variable X which has d bits assigned for decimal part and r bits assigned for the fractional part thus forming a number represented by $N = d + r$ bits. To account for negative numbers we assume two’s complement representation. The output distribution $f_Y(y)$ can be calculated using (1)

$$f_Y(y) = \sum_{k=0}^N P(k) f_Y^k(y) \quad (1)$$

Where $P(k)$ is the probability of having k bit flips simultaneously and can be calculated using:

$$P(k) = P_e^k (1 - P_e)^{N-k} \quad (2)$$

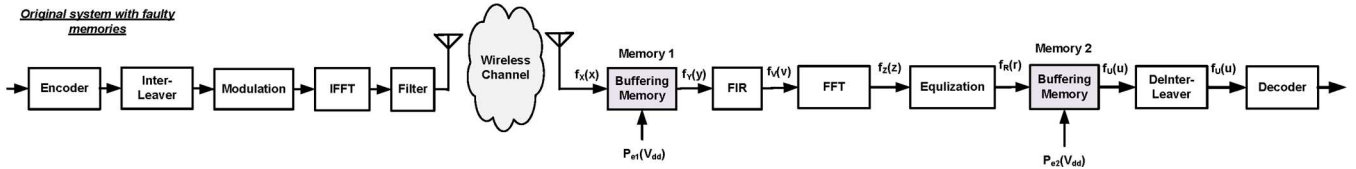


Fig. 4. A generic OFDM system with faulty memories.

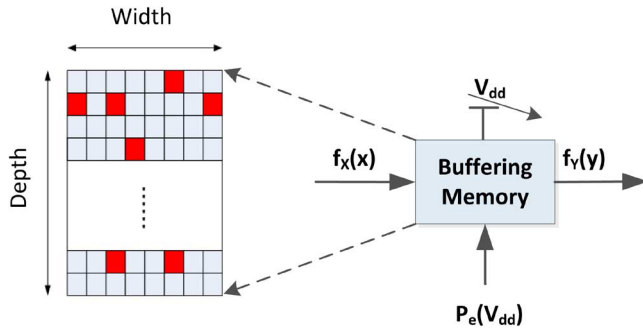


Fig. 5. Memory Failure model.

$f_Y^k(y)$ is the distribution of data when k bit flips occur at one word, where, k (number of bit flips) can be a number from 0 up to N

$$f_Y^k(y) = \frac{1}{k!} \sum_{n_1=0}^{N-1} \sum_{\substack{n_2=0 \\ n_2 \neq n_1}}^{N-1} \cdots \sum_{\substack{n_k=0 \\ n_k \neq n_1 \\ \dots n_k \neq n_{k-1}}}^{N-1} f_{Y_{n_1, n_2, \dots, n_k}}(y) \quad (3)$$

The interested reader is referred to [19] for a detailed analysis of this model.

B. Effect of Filtering on Memory Errors

Filtering is an integral part of any communication system, thus it is important to know how the data distribution changes after filtering. It is well known that filtering Gaussian distributed data will produce a new Gaussian distribution with a different mean and variance [25]. However, when VoS is applied to the buffering memory, the distribution of the retrieved data from the memory deviates from the Gaussian distribution. Hence, it is imperative to quantify the effect of filtering on the distribution of the retrieved data from the faulty buffering memory.

In the presence of faulty memories, the authors in [19] derived the distribution of the data after filtering based on the Fourier transform duality between the probability mass function (PMF) and the characteristic function [26].

In general, as shown in Fig. 6, given an FIR filter with an impulse response $h(k)$ and input data with a PMF $f_Y(y)$, the statistical PMF of the output data, $f_v(v)$, can be written as the convolution between multiple scaled versions of the input PMF, $f_Y(y)$ as follow:

$$f_V(v) = \frac{1}{\prod_k |h(k)|} f_Y\left(\frac{y}{h(0)}\right) * f_Y\left(\frac{y}{h(1)}\right) * \cdots \quad (4)$$

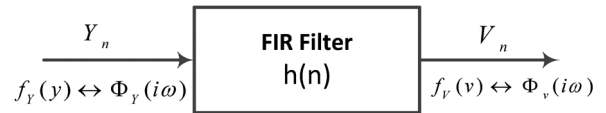


Fig. 6. FIR filtering.

C. Effect of FFT on Memory Errors

In most modern communication system orthogonal frequency division multiplexing is extensively used to combat channel fading, where the FFT is an integral block. Therefore, it is important to quantify how FFT affects the statistics of a complex sequence of data V for which the real and imaginary parts are independent with the same distribution. This may be the case when demodulating the received OFDM symbols, where the subcarriers have a certain distribution due to the effect on channel noise, memory errors and interference.

The real and the imaginary parts of the N -point FFT are given by:

$$Z_r(n) = \sum_{k=0}^{N_{FFT}-1} V_r(k) \cos\left(\frac{2\pi kn}{N_{FFT}}\right) + V_i(k) \sin\left(\frac{2\pi kn}{N_{FFT}}\right) \quad (5)$$

Similarly

$$Z_i(n) = \sum_{k=0}^{N_{FFT}-1} -V_r(k) \sin\left(\frac{2\pi kn}{N_{FFT}}\right) + V_i(k) \cos\left(\frac{2\pi kn}{N_{FFT}}\right) \quad (6)$$

We can express both the real and imaginary parts of the output as:

$$Z_i(n) = \sum_{k=0}^{N_{FFT}-1} P_n(k) + Q_n(k) \quad (7)$$

$$Z_r(n) = \sum_{k=0}^{N_{FFT}-1} S_n(k) + C_n(k) \quad (8)$$

We are interested in obtaining the distribution of both the real and imaginary parts of the output Z . Since the variables $S_n(k)$, $C_n(k)$, $P_n(k)$ and $Q_n(k)$ can be considered as random variables, the real and imaginary parts of the output of the FFT (Z_r and Z_i) can be derived as a sum of a large number of random variables, which by the central limit theory approaches an asymptotic Gaussian distribution. This means that the distribution of the data after the FFT can be approximated as a Gaussian distribution (with sufficiently large N_{FFT} [25], [27]). The authors in [20] have validated the Gaussian distribution of

the data after the FFT and derived an expression of the mean and the variance of the real and imaginary parts as follow:

$$\mu_{Zr} = \mu_{Zi} = \begin{cases} N_{FFT} \times \mu_v, & n = 0 \\ 0, & n = 1, 2, \dots, N_{FFT} - 1 \end{cases} \quad (9)$$

$$\sigma_{Zi}^2 = \sigma_{Zr}^2 = N_{FFT} \sigma_v^2 \quad (10)$$

If the distribution of the input data V has a zero mean ($\mu_v = 0$) which is the typical case for any wireless channel noise, then:

$$\mu_{Zr} = \mu_{Zi} = 0 \quad (11)$$

Hence, one can express the distribution of the data after the FFT in the system as a normal distribution $\mathcal{N}(0, N_{FFT} \sigma_v^2)$. In which the variance σ_v^2 of the data after the filtering can be obtained using the distribution $f_V(v)$ in (4) as follow:

$$\sigma_v^2 = E\{(v - \mu_v)^2\} = E\{v^2\} = \sum_v v^2 \times f_v(v) \quad (12)$$

D. Effect of Equalization on Data Distribution

For an OFDM system with a Rayleigh fading channel, the FFT stage converts the data distribution into Gaussian as discussed previously. Hence, the received signal for subcarrier k could be expressed as:

$$z_k = h_k s_k + \tilde{n}_k \quad (13)$$

where \tilde{n}_k is a complex Gaussian noise of zero mean and variance σ_n^2 which can be calculated by using (10) and average channel power $\bar{\gamma}$. The goal in this subsection is to find the distribution of the equalized signal r_k . Without loss of generality and for mathematical tractability, we assume a least squares equalization where one can express the equalized signal r_k as

$$r_k = s_k + \tilde{n}_k / h_k \quad (14)$$

The PMF of the equalized signal r_k can be obtained using given probability concept as described in the following equation

$$f_{R|s}(r|s) = \int_{\gamma=0}^{\gamma=\infty} f_{R|s,\gamma}(r|s,\gamma) \times f_{\gamma}(\gamma) d\gamma \quad (15)$$

where

$$f_{R|s,\gamma}(r|s,\gamma) \sim \mathcal{N}\left(s, \frac{\sigma_n^2}{|h_k|^2}\right)$$

and

$$\gamma = |h_k|^2, f_{\gamma}(\gamma) = \frac{1}{\bar{\gamma}} e^{-\frac{\gamma}{\bar{\gamma}}}$$

By using integration tables, the distribution of the equalized data can be given by:

$$f_{R|s}(r|s) = \frac{\sigma_n^2}{\bar{\gamma}} \left[\frac{2\sigma_n^2}{\bar{\gamma}} + (r - s)^2 \right]^{-3/2} \quad (16)$$

Then, by storing the equalized data r into another faulty buffering memory (memory 2) as shown in Fig. 4, the distribution of the data read from the memory $f_{U|s}(u|s)$ can be similarly derived as discussed in Section III-A.

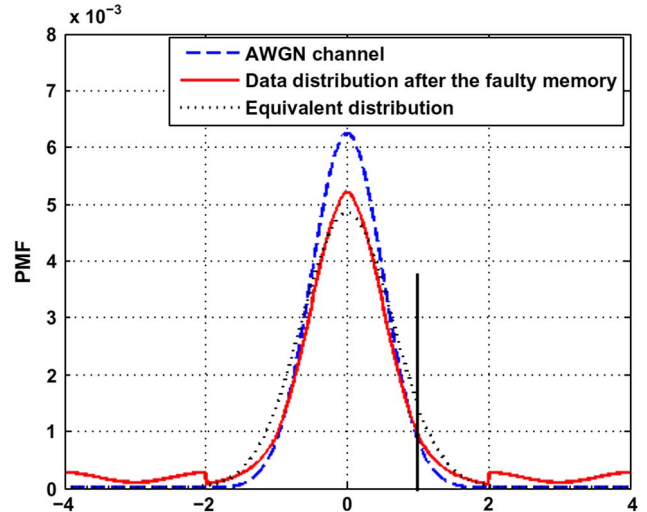


Fig. 7. Arbitrary distribution of the input data sequence.

The last block before the decoder is the de-interleaver which does not perform any computation on the data (mainly permutation and data shuffling). That is why the data distribution does not change after the deinterleaver. Furthermore, since the errors introduced within the memory are spatially uniform (not burst or cluster errors), the deinterleaver does not change the variance of the errors and hence does not change the performance quality of the decoder with regards to memory errors.

IV. EQUI-NOISE

As mentioned in Section II, one can model the hardware as an extension of the wireless channel in communication systems where quality is controlled by the operating conditions such as frequency and supply voltage. By propagating data statistics through various communication blocks, the area under the tail of the resultant distribution (after a certain threshold depending on the modulation) represents the BER. The key idea is to find an equivalent Gaussian noise distribution that has the same area under the tail of the distribution (or equivalently, the same BER) as the stat corrupted data statistics. Fig. 7 illustrates this concept. As shown in this figure, the distribution of the data affected by an AWGN channel stored in a faulty memory can be approximated by another Gaussian distribution that captures both AWGN and memory noise while assuming that the storing memory is fault-free. In other words, the area under the tail of both distributions after a given threshold is *equal* to BER.

A. BER of the System With Faulty Memory

For our analysis we assume BPSK modulation for simplicity. However, the same methodology could be applied to any other modulation scheme without any loss of generality. Considering the system with faulty memories shown in Fig. 4, the uncoded BER performance (before the FEC decoder) could be obtained by:

$$BER = P(s_1) \int_{-\infty}^0 f_{U|s_1}(u|s_1) du + P(s_2) \int_0^{\infty} f_{U|s_2}(u|s_2) du \quad (17)$$

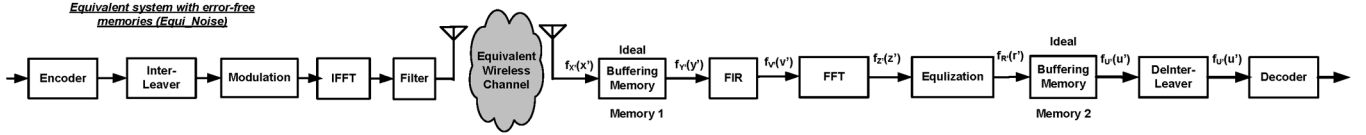


Fig. 8. Equi-noise system model.

Where $P(s_1)$ and $P(s_2)$ represents the probability of transmitting the BPSK symbols ($s_i = \pm 1, i = 1, 2$) and $f_{U|s_i}(u|s_i)$ is the distribution of the data before the decoder. Due to symmetry of the tails of the distributions $f_{U|s_1}(u|s_1)$ and $f_{U|s_2}(u|s_2)$ and assuming equally likely symbols, the BER is expressed in (18).

$$P(s_1) = P(s_2) = 0.5$$

$$\int_{-\infty}^0 f_{U|s_1}(u|s_1) du = \int_0^{\infty} f_{U|s_2}(u|s_2) du$$

$$BER = \int_0^{\infty} f_{U|s_2}(u|s_2) du \quad (18)$$

Hence, the BER is mathematically calculated based on the derived distribution, $f_{U|s_2}(u|s_2)$, which has been obtained by propagating the retrieved data distribution through the communication system blocks as presented in the previous section.

B. BER of the Equivalent System

Once the BER of the system with faulty memory is calculated, the goal is to find an equivalent noise, n_{eq} with zero mean and variance σ_{eq}^2 , such that the equivalent system with ideal buffering memories achieves the same BER performance of the original system with faulty buffering memories. The Gaussian distribution of the equivalent noise can be written as:

$$n_{eq} \sim f_{N_{eq}(n_{eq})} = \frac{1}{\sqrt{2\pi}\sigma_{eq}} e^{-\frac{n_{eq}^2}{2\sigma_{eq}^2}} \quad (19)$$

The target is to find a mathematical formula of the equivalent BER_{eq} which can be calculated using the PMF $f_{U'}(u')$ of the data before the decoder of the Equi-Noise system as shown in Fig. 8. Similarly, due to symmetry of the distribution tails and assuming equally likely symbols the BER is expressed by

$$BER_{eq} = \int_0^{\infty} f_{U'|s_2}(u'|s_2) du' \quad (20)$$

where $f_{U'|s_2}(u'|s_2)$ is obtained by propagating the equivalent noise distribution through the data path blocks (Memory, FIR, FFT and Equalizer). Since the memories of the equivalent system are ideal with no errors, the distribution of the data after the memory does not change. Due to the filter linearity, passing the equivalent noise n_{eq} through the filter stage will produce another Gaussian n_{eqFIR} with a zero mean and variance σ_{eqFIR}^2 given by:

$$\sigma_{eqFIR}^2 = \sum_k h(k)^2 \sigma_{eq}^2 \quad (21)$$

Similar to the discussion in Section III-C, since the equivalent noise after the filtering n_{eqFIR} has zero mean and variance σ_{eqFIR}^2 , the Gaussian noise after the FFT n_{eqFFT} will have zero mean and variance

$$\sigma_{eqFFT}^2 = N_{FFT} \times \sigma_{eqFIR}^2 \quad (22)$$

Thus, the received signal after the FFT for each subcarrier can be expressed as:

$$z'_k = h_k s_k + \tilde{n}_{eqFFT,k} \quad (23)$$

where

$$\tilde{n}_{eqFFT} \sim \mathcal{N}(0, \sigma_{eqFFT}^2)$$

After the ZF equalization:

$$r'_k = s_k + \tilde{n}_{eqFFT,k}/h_k \quad (24)$$

Following the derivation in Section III-D, the distribution of r'_k is written as

$$f_{R'|s}(r'|s) = \frac{\sigma_{eqFFT}^2}{\tilde{\gamma}} \left[\frac{2\sigma_{eqFFT}^2}{\tilde{\gamma}} + (r' - s)^2 \right]^{-3/2} \quad (25)$$

Since the second buffering memory and the interleaver do not change the data distribution, hence

$$f_{U'|s}(u'|s) = f_{R'|s}(r'|s)$$

Finally, after using integration table formula, the BER of the equivalent system described by (20) can be found as

$$BER_{eq} = \frac{1}{2} \times \left(1 - \frac{1}{\sqrt{1 + 2\sigma_{eqFFT}^2/\tilde{\gamma}}} \right) \quad (26)$$

By equating the mathematical formula of the BER_{eq} in (26) for the equivalent system and that of the original system with faulty memories in (18), the variance of the equivalent noise n_{eq} can be calculated as in (27).

$$\sigma_{eq}^2 = \frac{\tilde{\gamma}}{2N_{FFT} \times \sum_k h(k)^2} \left(\frac{1}{(1 - 2 \times BER)^2} - 1 \right) \quad (27)$$

It is worth mentioning that the previous analysis can be generalized for any other modulation scheme (QPSK, 16-QAM or 64-QAM), as well as for any number of memories in the system.

C. BER After FEC Decoder

Forward Error Correction (FEC) decoders are employed at the receiver to detect and correct channel errors. In this subsection we employ convolutional codes and use the Viterbi decoding algorithm at the receiver to decode the transmitted

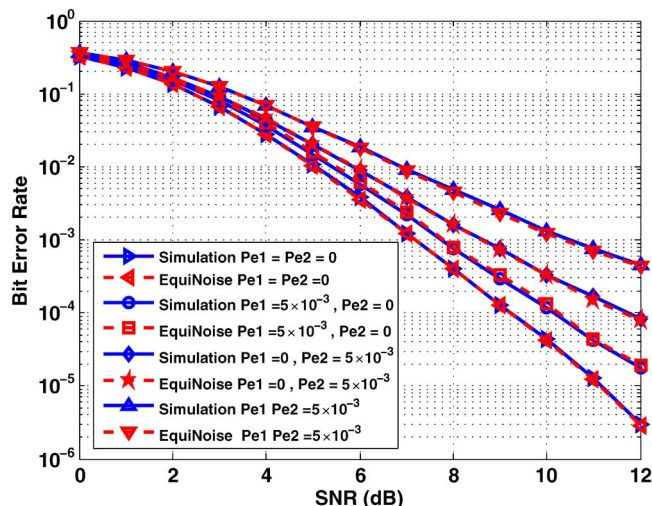


Fig. 9. Comparing the simulation results with the proposed model.

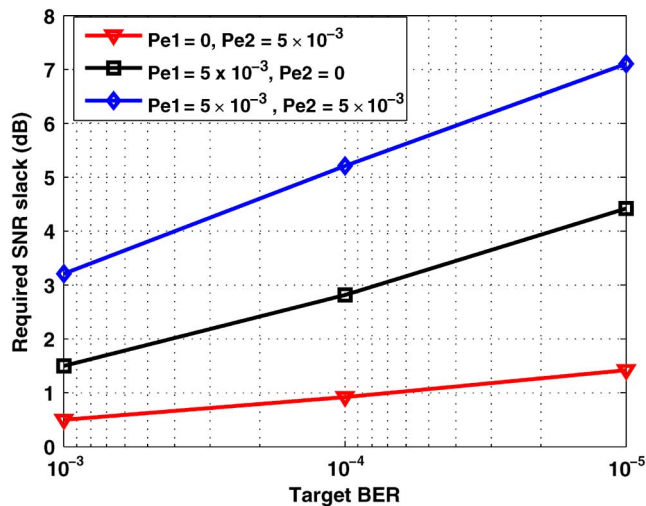


Fig. 10. The required SNR slack versus target BER of the equivalent noise system for different memory error rates.

bits. First, we discuss the equi-noise performance with the hard input FEC. Then, we extend our discussion to include soft-input Viterbi.

1) *Hard-Input FEC Decoder*: Hard-input FEC decoders employ hamming distance to find the branch metric distance. Since both the original and equivalent systems have the same BER before the decoder, both achieve the same coded-BER. Fig. 9 shows the BER performance after the Viterbi decoder for both systems in Fig. 4 and Fig. 8 with different memory error rates. Note the close match between the simulations results of the original system with faulty systems and these of the equi-noise.

Fig. 10 shows the SNR slack versus the target BER for different combination of memory error rates. The memory errors manifest as a reduction in the received SNR where a higher slack in the received SNR is required to achieve the target BER. It is important to note that the memory error effects on the SNR depend on the location of the memory as will be explained in the next section.

2) *Soft-Input FEC Decoder*: The conventional soft-input Viterbi algorithm is based on the Maximum Likelihood (ML) criteria assuming Gaussian noise. However, incorporating

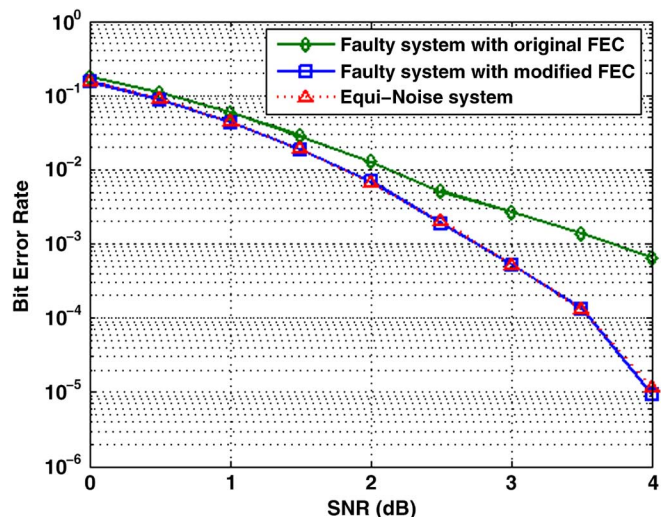


Fig. 11. Equivalent noise with soft-input FEC in AWGN channel.

faulty memories will result in a new distribution that is slightly non-Gaussian [19], [20]. Thus, the performance of the conventional Viterbi decoder in the presence of memory errors suffers some degradation. Hence, by applying the equivalence of the BER before the Viterbi decoder between the faulty system and the equi-noise system, the equivalence of the BER after the conventional FEC decoder is not achieved. Fig. 11 shows that the BER performance of the faulty system with original FEC is not the same as the equi-noise system.

This problem is addressed by incorporating the actual statistics of the data after faulty memories while calculating the log likelihood ratios (LLR) as described by prior work of the authors in [28], which presents a modified FEC decoder that is based on the ML criteria. Fig. 12 validates this approach where a very close match exists between the BER performance of the modified FEC for the system with faulty memories and that for the equivalent system. It is worth mentioning that for very high error rates in the memory ($P_e > 10^{-2}$) a divergence of less than 0.27 dB occurs at a target BER of 10^{-5} . For more realistic values of memory error ($P_e > 10^{-4}$), the model divergence is less than 0.01 dB.

The work in [28] was expanded to include many different families of FEC such as Turbo decoders as well as Low Density Parity Codes [29], [30].

V. MEMORY SENSITIVITY ANALYSIS

In typical communication systems, buffering memories are needed to store the data before and after processing by basic blocks such as FFT, channel estimation, interleaver and equalization. These memories differ in size and the level of the data redundancy. Generally speaking, one would expect that the closer the buffering memory is to the decoder, the lower the data redundancy level, however, as we will discuss later, this is not necessarily always the case. Depending on its location in the processing chain, each block affects system performance in different ways. It is therefore interesting to evaluate the impact of each buffering memory on the system quality of service (QoS) measured by the BER. A straightforward way

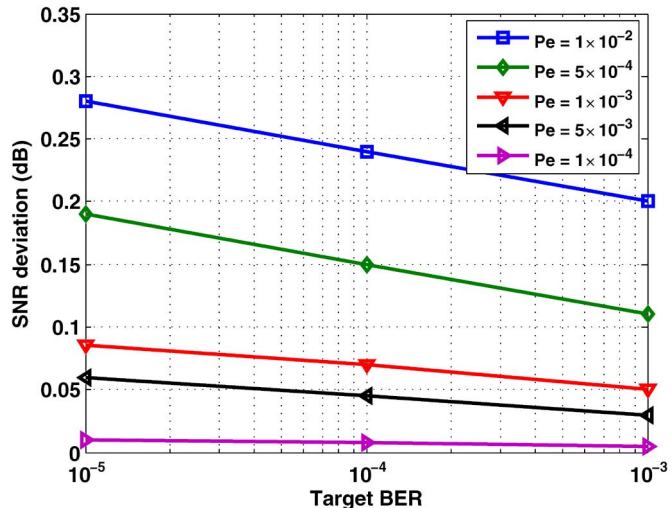


Fig. 12. Model accuracy for different memory error rates and target BER's with soft-input FEC.

to address this problem is by performing a sensitivity analysis of the system's BER with respect to the amount of error rate applied at each memory.

We consider two buffering memories. The first one (M_1) is the buffering memory at the receiver front end immediately after the analog to digital conversion. While the second memory (M_2), is the memory preceding the FEC decoder. It is worth mentioning that the memory word length N has an impact over the possible error value introduced to the word. Memories with large word length will have a higher probability that the retrieved word is erroneous as compared to memories with smaller word length. The sensitivity of the BER to the probability of error is defined as:

$$S_{P_e}^{BER} = \frac{\frac{\partial BER}{\partial P_e}}{\frac{BER}{P_e}} \quad (28)$$

where the BER is given by

$$BER = \int_0^{\infty} f_v(v, P_e) dv \quad (29)$$

and $f_v(v, P_e)$ is the data distribution before the decoder which depends on the error rate applied at the buffering memory under consideration.

$$\frac{\partial BER}{\partial P_e} = \int_0^{\infty} \frac{\partial f_v(v, P_e)}{\partial P_e} dv \quad (30)$$

where the derivative can be approximated by

$$\frac{\partial f_v(v)}{\partial P_e} \approx \frac{f_v(v, P_e + \Delta P_e) - f_v(v, P_e)}{\Delta P_e} \quad (31)$$

Fig. 13 illustrates the sensitivity of the BER with respect to an error rate of 5×10^{-3} in each memory, assuming an un-coded system. The first observation is that as the SNR increases, the

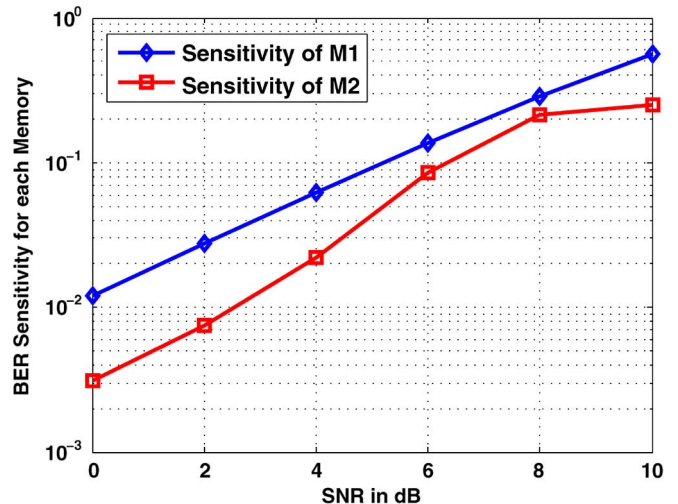


Fig. 13. Sensitivity of BER at error rate 5×10^{-3} .

system performance is more sensitive to error rate in the memories. This is expected since at higher SNR, the effect of channel noise is less as compared to the errors from the memory.

The second observation is that, for the same error rate, M_1 and M_2 impact the system differently. Interestingly, contrary to what it was expected, system performance is more sensitive to errors in M_1 than M_2 although M_2 is closer to the decoder as compared to M_1 . The reason behind that is the FFT, since it is not a one-to-one mapping process. Any error at one of the inputs of the FFT will affect all the N_{FFT} data at the output of the FFT. Therefore, errors in M_1 have a more severe effect on system performance, especially for larger FFT sizes.

VI. POWER MANAGEMENT POLICY

Based on the equi-noise modeling presented in the previous sections, the effect of memory supply voltage over scaling (VOS) on the final metric of the system (BER in this case) can be mathematically derived. For the different combinations of the supply voltages, Equi-Noise provides a mathematical model that precisely estimates system performance at any given SNR as a result of the applied power management technique. We considered a system with two buffering memories. The target of the algorithms is to find the appropriate memory supply voltage that maximizes power savings by exploiting the available SNR slack while keeping the system performance within the required margin.

The supply voltages and the equivalent memory error rates shown in Table II are based on HSPICE circuit simulations of a 6 T SRAM cells using 65 nm CMOS predictive technology model [31]. Different SRAM memories could have different memory error rates due to process variations. Furthermore, due to aging and temperature variations, memory error rates may vary. Hence, Built-in self-test (BIST) mechanisms could be applied to measure and characterize memory error rates under VOS. The power manager algorithm will then run the BIST technique for each memory to update the entries of the memory

TABLE II
 SUPPLY VOLTAGE AND CORRESPONDING MEMORY ERROR RATE

V_{dd}	1.0	0.85	0.75	0.65
P_e	$1.69e^{-15}$	$5.21e^{-12}$	$1.8e^{-6}$	$2.7e^{-3}$

error rates for different supply voltages. Since temperature variations and other factors that affect the memories are slow processes, once the table is constructed, it will need infrequent updates, with negligible overhead and throughput degradation.

Algorithm 1: Offline Power Manager Algorithm

```

1: for each quantized SNR do
2:    $V_{dd} = \{1.0, 0.85, 0.75, 0.65\}$ 
3:   for  $V_1 = \{1.0, 0.85, 0.75, 0.65\}$ 
4:     for  $V_2 = \{1.0, 0.85, 0.75, 0.65\}$ 
5:        $P_{e1} = LUT(V_{Mem1});$ 
6:        $P_{e2} = LUT(V_{Mem2});$ 
7:       Calculate  $BER_{uncoded}(P_{e1}, P_{e2}, SNR_q);$ 
8:       Calculate  $\sigma_{eq^2}$  (Equi-Noise model);
9:       Calculate the effective  $SNR_{eff};$ 
10:      Calculate Power Savings  $P_s(P_{e1}, P_{e2});$ 
11:    end for
12:  end for
13: end for
    
```

The proposed power management technique is composed of two parts. The first part is an offline algorithm that characterizes the effective SNR of the system based on the memory error rates and received SNR. In more details, for every pair of memory supply voltages (V_1, V_2), the algorithm reads the corresponding memory error rates (P_{e1}, P_{e2}). Then based on the derived PMF distribution under VOS, the equivalent noise that achieves same BER under these supply voltages is calculated. Hence, the value of the effective SNR is tabulated for the tuple of (SNR_{rec}, V_1, V_2). The details of this algorithm are explained in Algorithm 1.

The size of the LUT depends on the quantization resolution of the received SNR and the number of the allowed memory supply voltages. Assuming a linear quantization of the received SNR with a step ΔSNR and 12-bit precision to store the effective SNR, the size of one LUT for a certain combination of (V_1, V_2) is given by

$$LUT_{size} = 12 \times \frac{SNR_{range}}{\Delta SNR} bits$$

Hence, the total required storage for the different combination of the two memories supply voltages can be expressed as:

$$N_{mem_{volt}}^2 \times LUT_{size}$$

Fig. 14 shows the required memory storage of the LUT versus the resolution of the ΔSNR step. As expected, a lower quantization resolution of the received SNR will result into a large storage requirement. On the other hand, this quantization will result into a reduced accuracy of the algorithm where a range of the received SNR's are mapped into one quantized effective SNR. Fig. 15 shows the mapping between the received SNR and

the effective SNR under different memory error rates. It is clear that for a quantization error of $\Delta SNR/2$ in the received SNR, the maximum error of the effective SNR is also of $\Delta SNR/2$ which occurs at small error rates in the memories or at lower values of the received SNR. To reflect this quantization error in the effective SNR to the accuracy of the calculated BER, Fig. 16 shows the maximum ratio of the LUT-based BER to the exact BER without quantization versus the SNR step resolution. Based on this analysis, the quantization step is chosen to be 0.02 dB to have a considerable small LUT and a maximum error rate less than 10% as seen in Fig. 16. Another pessimistic approach to alleviate this error can be applied where a safety margin of 0.01 dB (half the SNR step resolution) can be subtracted from the tabulated effective SNR. It is worth mentioning here that the number of bits used to represent the value of the estimated received SNR could reduce the table size. In general, step size of the LUT (ΔSNR) should not be smaller than the accuracy of the fixed point representation of the estimated SNR 2^{-N_f} , where N_f is the number of fractional bits. If the estimated received SNR has less accuracy (lower number of bits) than the SNR step (ΔSNR), then the table size can be smaller.

$$\Delta SNR \geq 2^{-N_f}$$

Algorithm 2: Online Power Manager Algorithm

```

1: for each time step do
2:   Initialize  $V_{Mem1} = V_{Mem2} = V_{dd,nominal}, P_{s,Mem} = 0$ 
3:   Estimate channel parameters:  $\sigma_n^2, \bar{\gamma}$ 
4:   Quantize received SNR,  $SNR_{rec,q} = Q(SNR_{rec})$ 
5:   Calculate available slack  $\Delta SNR_{Av} = SNR_{rec,q} - SNR_{req}$ 
6:   if  $\Delta SNR_{Av} > 0$ 
7:     for  $V_{Mem1} = \{1.0, 0.85, 0.75, 0.65\}$ 
8:       for  $V_{Mem2} = \{1.0, 0.85, 0.75, 0.65\}$ 
9:          $SNR_{eff}(v_1, v_2) = LUT(v_1, v_2, SNR_{rec,q});$ 
10:         $P_{s,Mem} = P_s(v_1, v_2);$ 
11:        if ( $SNR_{eff} > SNR_{rec,q}$ ) & ( $P_s > P_{s,Mem}$ )
12:           $P_{s,Mem} = P_s;$ 
13:          Update candidate  $V_{dd} = (v_1, v_2)$ 
14:        end if
15:      end for
16:    end for
17: end for
    
```

The second part of the algorithm is the online part, which updates the memory supply voltages to track the channel SNR variations. The online algorithm shown in Algorithm 2 runs every time step (ΔT) which is chosen to be smaller than the coherence time of the channel ($\Delta T < T_{coherence}$). Initially, it estimates the channel parameters and calculates the average received SNR. Based on the target BER performance of the system, a target SNR_{req} is required to achieve the BER requirement. Hence, the available slack in the SNR (ΔSNR) is calculated. The algorithm loops over the different combinations of quantized memory supply voltages, then reads the effective SNR from the LUT's and the corresponding power savings (steps 9–10). The combination of the memories supply voltages that has the highest power savings and achieves the target

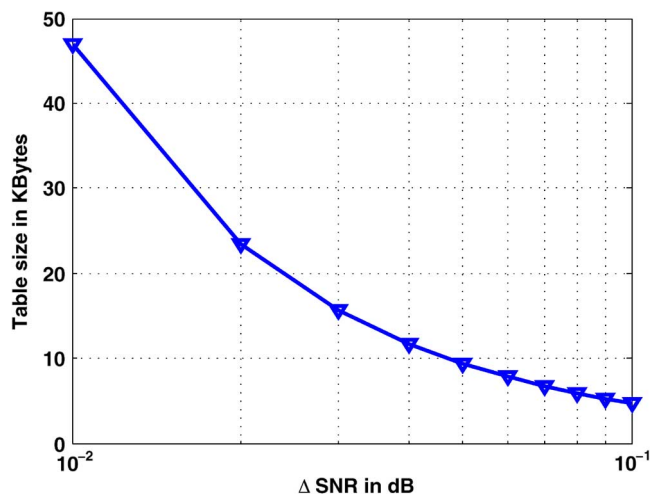


Fig. 14. Size of LUT versus the SNR quantization step (ΔSNR).

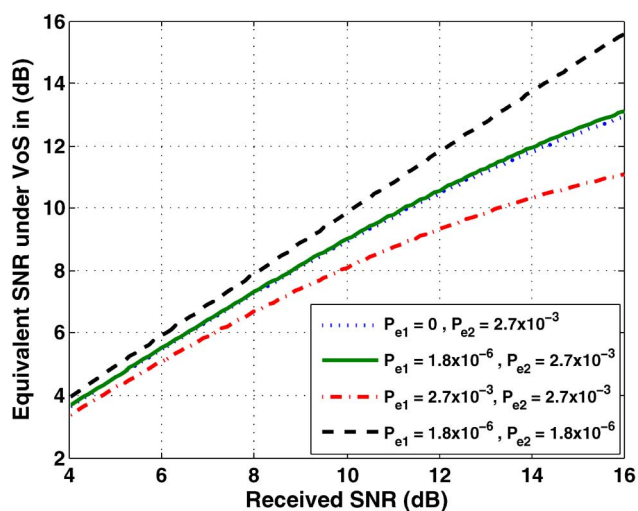


Fig. 15. Effective SNR versus received SNR for different memory error rates.

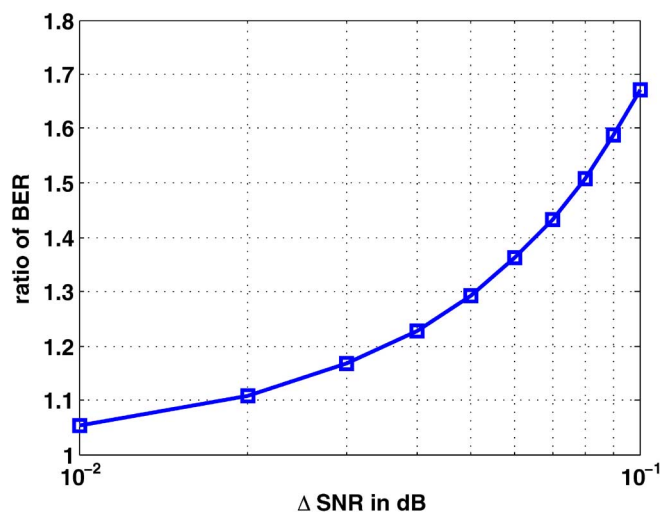


Fig. 16. BER deviation versus the SNR quantization.

BER performance is then chosen by the power manager for the buffering memories (steps 11–13).

VII. CASE STUDY

To fully evaluate the proposed approach, we applied the methodology to a DVB-T system as shown in Fig. 17. At the transmitter, the input transport stream is applied to the outer coder, shortened Reed-Solomon (RS 204,188) followed by a convolution interleaver. The second step of channel coding is applied via a puncturing convolutional coder which is followed by the inner DVB interleaver. The mapper module uses three constellation schemes: QPSK, 16QAM and 64QAM. The OFDM module conducts IFFT operation to transform 2k or 8k mode symbols into time domain symbols.

The channel is modeled as a Rayleigh fading channel. At the receiver, the incoming complex data is transformed again to the frequency domain through the FFT operation, followed by inner and outer channel decoding as shown in Fig. 17. We simulated the system with a QPSK modulation scheme with 2048 FFT and 3/4 punctured convolutional code.

We target two of the largest memories in the system, the buffering and FFT memory after the ADC ($M1$) and the buffering memory after the FFT ($M2$) which is used for the channel equalization. As discussed in the previous section, memories before the FFT have a higher impact on the system overall BER. In [13] a COFDM Baseband Receiver for DVB-T/H applications was implemented in 0.18 μm technology. In this implementation, 158 Kbytes of embedded buffering memories were required for FFT processing, channel estimation and equalization. These memories occupy approximately around 35% of the chip area and consume 25% of the total chip power. The area and power consumption for both the memories and logic were scaled down to 65 nm as shown in Table III to quantify for process variation at advanced CMOS technology. Note that in other more advanced schemes such as LTE etc., the memory share of area and power are more pronounced due to buffering requirements of techniques such as HARQ and MIMO etc. It is also important to note that in such advanced systems, typically advanced modulation and coding are used (AMC), thus two loops will need to be jointly optimized. The slower outer AMC loop controls the slack seen by the receiver, and the faster inner loop of the power manager minimizes power based on available slack.

A. Simulation Results

To verify the accuracy of the proposed model, we performed a full MATLAB simulation of the system depicted in Fig. 17 and compared the simulation BER to that generated by the proposed analytical model. It is important to note that simulating the whole DVB system having only two faulty memories at different probability of failures will take several days depending on the simulation engine used, whereas using the Equi-noise model, the BER can be calculated in seconds. Fig. 18 shows the BER performance after the soft-input Viterbi for the DVB system obtained from both the simulation and the Equi-noise model. As shown, there is close match between the simulation and the results using the proposed equi-noise model. The difference between the results obtained by using the proposed method and simulation depends on error rates in both memories as discussed previously in Section IV-C. However, this mismatch can be alleviated. Since the BER of the equi-noise is optimistic by a

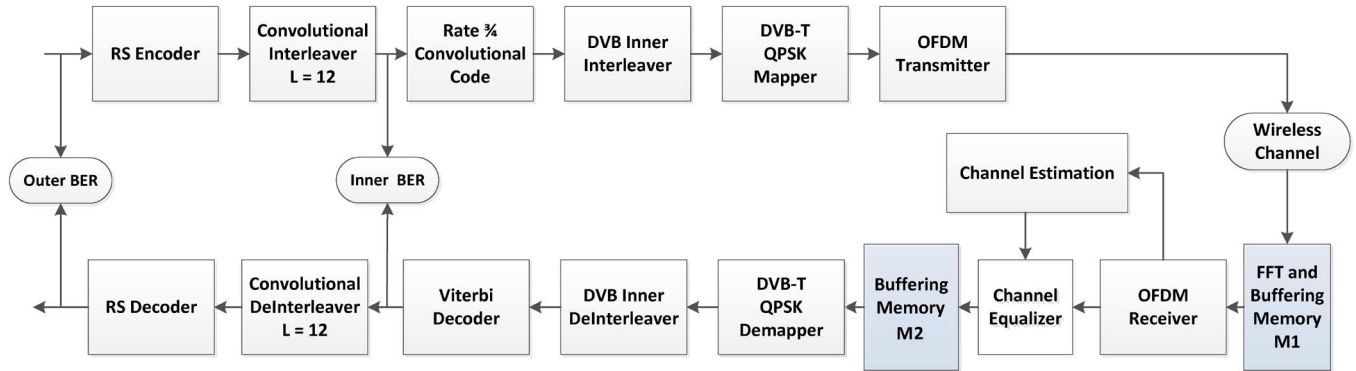


Fig. 17. DVB tranreceiver block diagram.

 TABLE III
 AREA AND POWER SHARES FOR MEMORY AND LOGIC[13]

	Count	Area	Power 0.18 μ m	Power 65 nm
Memory	158K Bytes	40.10%	65.6225 mW	7.067 mW
Logic	317 K Gates	59.90%	187.87 mW	21.2 mW

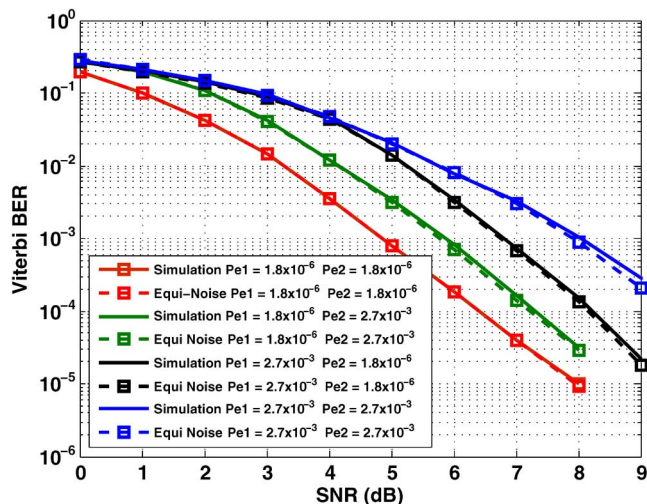


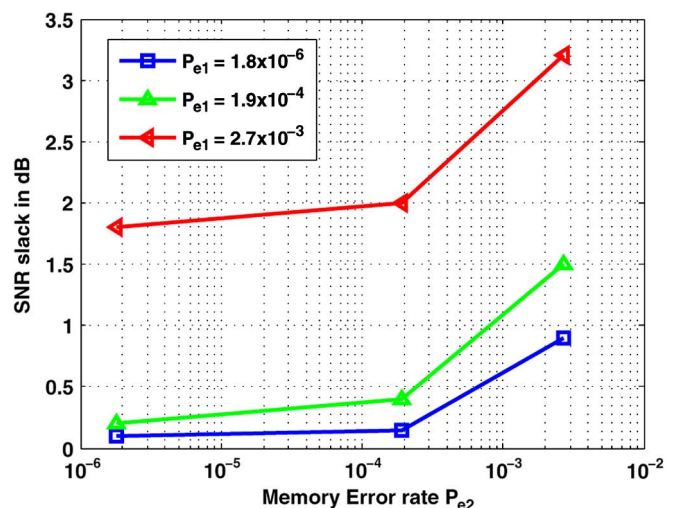
Fig. 18. Comparison between simulation and “Equi-noise” BER performance for the DVB system with different values of memory error rates.

fractional of dB, by adding a correcting factor (back-off factor) equivalent to that mismatch to the estimated SNR, the equivalence can be achieved.

Based on the equivalent noise model, Fig. 19 shows the required SNR slack for a target BER of 10^{-4} with different error rates of the two large memories in the system. When both memories are operating at a low P_e (10^{-6}), the effective SNR is slightly less than the received SNR. That is why, a very small SNR slack is required to achieve the target BER. However as the memories error rates increase, the noise floor will increase accordingly. Hence a higher slack in the received SNR is required which depends on the location of each memory, the corresponding memory error rates and the sensitivity of the final BER to that specific memory. The corresponding power saving achieved via voltage over-scaling for each memory for the coded DVB is given in Table IV.

 TABLE IV
 MEMORY POWER SAVINGS VERSUS SUPPLY VOLTAGE

Power Savings (%)	$V_1 = 1.0$	$V_1 = 0.85$	$V_1 = 0.75$	$V_1 = 0.65$
$V_2 = 1.00$	0%	16.68%	28.82%	39.4%
$V_2 = 0.85$	5.27%	22.15%	34.09%	44.67%
$V_2 = 0.75$	9.101%	25.78%	37.92%	48.50%
$V_2 = 0.65$	12.45%	29.14%	41.27%	51.85%


 Fig. 19. Required SNR slack to achieve 10^{-4} target BER for different combination of memory error rates.

B. Power Manager

As the wireless channel changes, the quality of the received signal in terms of SNR, exhibits time varying behavior. The power manager keeps track of the average received SNR and exploits the available slack to reduce the supply voltage of the buffering memories as explained in Algorithm-1 and Algorithm-2. By employing the power manager in the DVB system to modulate the supply voltage of the buffering memories, the power savings depends on the available slack in the SNR. Fig. 20 shows a sample of an SNR trace for a 5-tap Rayleigh fading channel with QPSK transmitted symbols for a target BER of 10^{-4} after the Viterbi. The effective SNR along with the variation of the supply voltage for both memories versus time is shown in Fig. 20. It is clear that the more the slack in the SNR, the higher the savings are. Fig. 21 shows the total

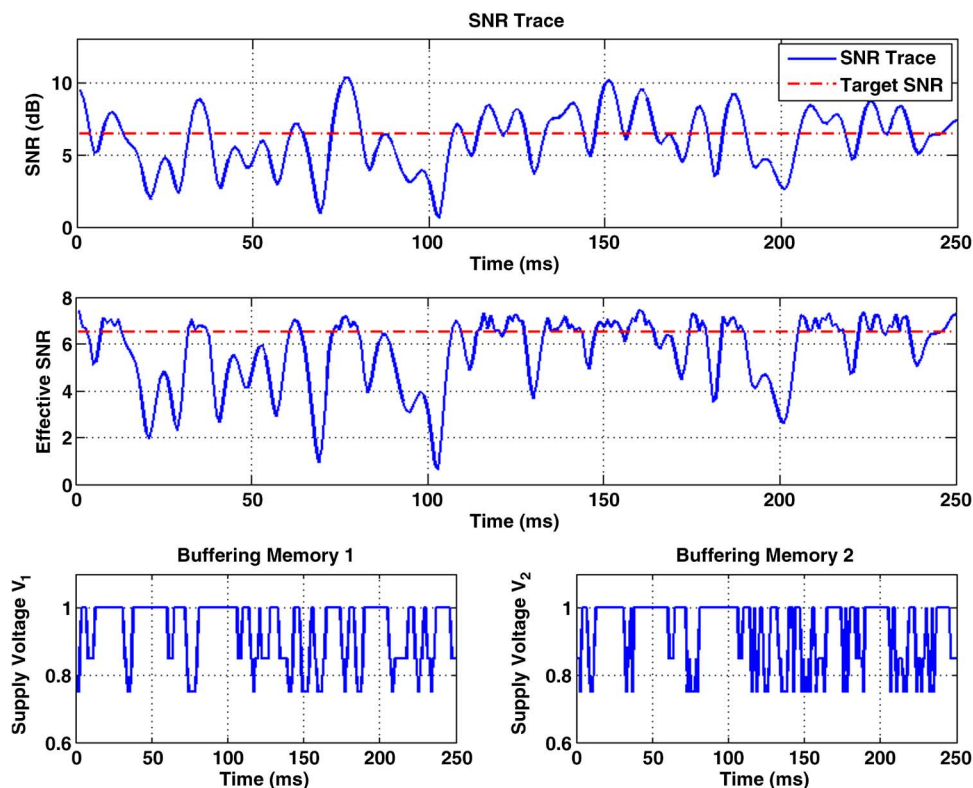


Fig. 20. SNR trace, effective SNR and the corresponding memory supply voltages for the DVB system based on Equi-Noise framework.

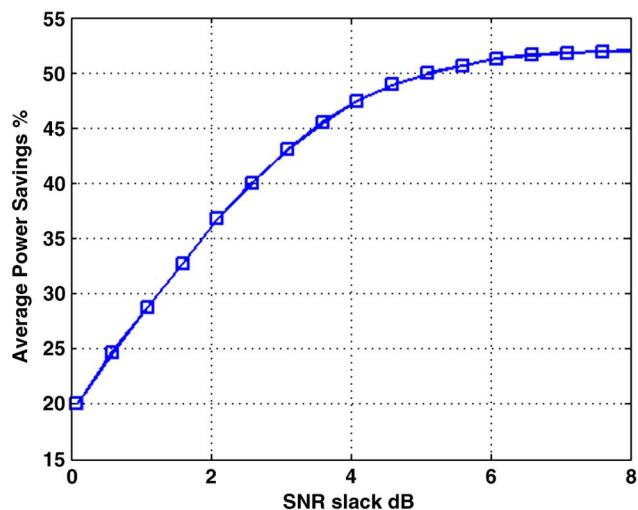


Fig. 21. Average Power savings of memories versus SNR slack.

average power savings for both memories versus the average SNR slack.

VIII. CONCLUSION

This paper proposed a statistical model to accurately and rapidly evaluate the impact of memory failures due to voltage over-scaling as a means of power management. The effect on the data distribution at each block in the communication system leading to and originating from the memory in question is quantified in a closed form solution. By replacing the traditional noise model in communication systems with the developed “Equi-Noise” model, one can investigate different power management policies, where the faulty hardware can

be treated as error-free hardware. The accuracy of the model is verified by performing a full simulation of a DVB system which demonstrates that results from the simulation are in close agreement with those obtained by the proposed analytical methods.

ACKNOWLEDGMENT

The authors would like to thank K. Amiri for his valuable discussions on the equi-noise model.

REFERENCES

- [1] International Technology Roadmap for Semiconductors [Online]. Available: <http://www.itrs.net>
- [2] B. H. Calhoun and A. P. Chandrakasan, “Ultra-dynamic voltage scaling (UDVS) using sub-threshold operation and local voltage dithering,” *IEEE J. Solid-State Circuits*, vol. 41, no. 1, pp. 238–245, Jan. 2006.
- [3] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, “Statistical design and optimization of SRAM cell for yield enhancement,” in *Proc. IEEE/ACM Int. Conf. Computer Aided Design*, Nov. 2004, pp. 10–13.
- [4] S. Das, D. Roberts, L. Seokwoo, S. Pant, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, “A self-tuning DVS processor using delay-error detection and correction,” *IEEE J. Solid-State Circuits*, vol. 41, no. 4, pp. 792–804, Apr. 2006.
- [5] F. J. Kurdahi, A. Eltawil, K. Yi, S. Cheng, and A. Khajeh, “Low-power multimedia system design by aggressive voltage scaling,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 18, pp. 852–856, 2010.
- [6] G. Karakonstantis, C. Roth, C. Benkeser, and A. Burg, “On the exploitation of the inherent error resilience of wireless systems under unreliable silicon,” in *Proc. 49th Annual Design Automation Conf.*, 2012, pp. 510–515, ACM.
- [7] C. Brehm, M. May, C. Gimmler, and N. Wehn, “A case study on error resilient architectures for wireless communication,” in *Architecture of Computing Systems*. New York, NY, USA: Springer, 2012, pp. 13–24.
- [8] F. Kurdahi, A. Eltawil, A. K. Djahromi, M. Makhzan, and S. Cheng, “Error-aware design,” in *Proc. 10th Euromicro Conf. Digital System Design Architectures, Meth. Tools*, Aug. 29–31, 2007, pp. 8–15.

- [9] Y. Liu, T. Zhang, and K. K. Parham, "Analysis of voltage over scaled computer arithmetic's in low power signal processing systems," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, Oct. 26–29, 2008, pp. 2093–2097.
- [10] A. K. Djahromi, S. Cheng, A. M. Eltawil, and F. J. Kurdahi, "Power management for cognitive radio platforms," in *IEEE Global Telecommun. Conf.*, Nov. 26–30, 2007, pp. 4066–4070.
- [11] Digital Video Broadcasting (DVB): Frame Structure, Channel Coding and Modulation for Digital Terrestrial Television (DVB-T) ETSI, 2004, Tech. Rep. ETSI EN 300 744.
- [12] C. Chi-Chie, S. Chi-Hong, and W. Jen-Ming, "A low power baseband OFDM receiver IC for fixed WiMAX communication," in *Proc. IEEE Asian Solid-State Circuits Conf.*, Nov. 12–14, 2007, pp. 292–295.
- [13] C. Lei-Fone, C. Yuan, C. Lu-Chung, M. Ying-Hao, L. Chia-Hao, L. Yu-Wei, L. Chien-Ching, L. Hsuan-Yu, H. Terng-Yin, and L. Chen-Yi, "A 1.8 V 250 mW COFDM baseband receiver for DVB-T/H applications," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 6–9, 2006, pp. 1002–1011.
- [14] *IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, IEEE 802.16, 2004.
- [15] A. Nilsson, E. Tell, and D. Liu, "An 11 mm², 70 mW fully programmable baseband processor for mobile WiMAX and DVB-T/H in 0.12 μ m CMOS," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 90–97, Jan. 2009.
- [16] A. B. Ericsson, Long term evolution (LTE): An introduction White Paper, October 2007.
- [17] R. Hegde and N. R. Shanghai, "A voltage over scaled low-power digital filter IC," *IEEE J. Solid-State Circuits*, vol. 39, no. 2, pp. 388–391, Feb. 2004.
- [18] A. K. Djahromi, A. M. Eltawil, and F. J. Kurdahi, "Fault tolerant approaches targeting ultra low power communications system design," in *Proc. IEEE Vehicular Technol. Conf.*, April 22–25, 2007, pp. 2600–2604.
- [19] A. Khajeh, K. Amir, M. S. Khairy, A. M. Eltawil, and F. J. Kurdahi, "A unified hardware and channel noise model for communication systems," in *Proc. IEEE Global Telecommun. Conf.*, Dec. 6–10, 2010, pp. 1–5.
- [20] M. S. Khairy, A. Khajeh, A. M. Eltawil, and F. J. Kurdahi, "FFT processing through faulty memories in OFDM based systems," in *Proc. IEEE GLOBECOM Workshops Appl. Commun. Theory Emerging Memory Technol.*, Dec. 6–10, 2010, pp. 1946–1951.
- [21] C. Roth, C. Benkeser, C. Studer, G. Karakonstantis, and A. Burg, Data Mapping for Unreliable Memories arXiv preprint arXiv:1212.4950, 2012.
- [22] A. K. Djahromi, A. M. Eltawil, F. J. Kurdahi, and R. Kanj, "Cross layer error exploitation for aggressive voltage scaling," in *Proc. Int. Symp. Quality Electronic Design*, March 26–28, 2007, pp. 192–197.
- [23] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS," *IEEE Trans. Computer-Aided Des. Integr. Circuits Syst.*, vol. 24, no. 12, pp. 1859–1880, Dec. 2005.
- [24] A. Khajeh, A. M. Eltawil, and F. J. Kurdahi, "Embedded memories fault-tolerant pre and post silicon optimization," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 19, no. 10, pp. 1916–1921, Oct. 2011.
- [25] A. Papulis, *Probability, Random Variables and Stochastic Processes*. New York, NY, USA: McGraw-Hill, 1965.
- [26] J. Brown and H. Piper, "Output characteristic function for an analog crosscorrelator with band pass inputs," *IEEE Trans. Inf. Theory*, pp. 6–10, Jan. 1967.
- [27] J. Schoukens and J. Renneboog, "Modeling the noise influence on the Fourier coefficients after a discrete Fourier transform," *IEEE Trans. Instrum. Meas.*, vol. IM-35, pp. 278–286, 1986.
- [28] A. Hussien, M. S. Khairy, A. Khajeh, A. M. Eltawil, and F. J. Kurdahi, "Combined channel and hardware noise resilient viterbi decoder," in *Proc. Asilomar Conf. Signals, Systems, Comput.*, pp. 395–399, 7–10 Nov.
- [29] A. M. A. Hussien, M. S. Khairy, A. Khajeh, A. M. Eltawil, and F. J. Kurdahi, "A class of low power error compensation iterative decoders," in *Proc. IEEE Global Telecommun. Conf.*, Dec. 5–9, 2011, pp. 1–6.
- [30] J. Geldmacher and J. Gotze, "On fault tolerant decoding of Turbo codes," in *Proc. 7th Int. Symp. Turbo Codes and Iterative Information Processing (ISTC)*, Aug. 2012, pp. 245–249.
- [31] Predictive Technology Model (PTM) [Online]. Available: <http://www.eas.asu.edu/~ptm>



digital signal processing, algorithm design and VLSI architecture for wireless communication systems.



power SoC design, design of low power high performance circuits for communication and multimedia applications, cross-layer optimization, fault tolerant adaptation, high performance high yield memory design, and low power DSP design. He was with Qualcomm low power DSP team from 2010 to 2012 researching and implementing advance low power technique for DSP cores for wireless multimedia applications. He is currently a research scientist at SoC Design Lab at Intel Labs researching ultra-low power SoC design, low power integration methods, and near threshold voltage design.



Dr. Eltawil received several distinguished awards, including the NSF CAREER award in 2010 supporting his research in low power systems.



Dr. Kurdahi was Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: ANALOG AND DIGITAL SIGNAL PROCESSING from 1993 to 1995, Area Editor in IEEE DESIGN AND TEST for reconfigurable computing, and served as program chair, general chair or on program committees of several workshops, symposia and conferences in the area of CAD, VLSI, and system design. He received the best paper award for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS in 2002, the best paper award in 2006 at ISQED, and other distinguished paper awards at DAC, EuroDAC and ASP-DAC.

Muhammad S. Khairy received the B.Sc. and M.Sc. degrees (hons) from the Electronics and Communications Department, Cairo University, Egypt in 2005 and 2009, respectively. He is currently working towards the Ph.D. degree at the EECSS Department, University of California, Irvine, CA, USA.

He worked as a Research Intern at the Institute of Communications and Radio-Frequency Engineering at Vienna University of Technology, Austria in 2009. His research interest includes wireless communication, low power design of digital communication, algorithm design and VLSI architecture for wireless communication systems.

Amin Khajeh (S'01–M'11) received the B.Sc. degree in electrical engineering and Communication from Shiraz University, Iran, in 2002 and the M.Sc. degree in electrical engineering from University of Texas at Arlington, TX, USA, in 2005, and the Ph.D. degree in EECS from University of California, Irvine, CA, USA, in 2010.

He was an intern in research and development division of Siemens Company in summer of 2002 and he later joined Siemens as a research staff member from 2002 to 2003. His research interests include low power SoC design, design of low power high performance circuits for communication and multimedia applications, cross-layer optimization, fault tolerant adaptation, high performance high yield memory design, and low power DSP design. He was with Qualcomm low power DSP team from 2010 to 2012 researching and implementing advance low power technique for DSP cores for wireless multimedia applications. He is currently a research scientist at SoC Design Lab at Intel Labs researching ultra-low power SoC design, low power integration methods, and near threshold voltage design.

Ahmed M. Eltawil (S'97–M'03) received the Ph.D. degree from the University of California, Los Angeles, CA, USA, in 2003.

Since 2005, he has been with the Department of Electrical Engineering and Computer Science, University of California, Irvine, where he is currently an Associate Professor. He is the founder and director of the Wireless Systems and Circuits Laboratory. His current research interests are in low power digital circuit and signal processing architectures for wireless communication systems.

Fadi J. Kurdahi received the Ph.D. degree from the University of Southern California, CA, USA, in 1987.

Since then, he has been a member of the faculty at the Department of Electrical and Computer Engineering at UCI, where he conducts research in the areas of Computer Aided Design of VLSI circuits, high-level synthesis, and design methodology of large scale systems, and serves as the Associate Director for the Center for Embedded Computer Systems (CECS).