This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Fault-Tolerant Embedded-Memory Strategy for Baseband Signal Processing Systems

Vadim Smolyakov, *Student Member, IEEE*, Glenn Gulak, *Senior Member, IEEE*, Timothy Gallagher, *Member, IEEE*, and Curtis Ling, *Senior Member, IEEE*

***Abstract*— The growing density of integration and the increasing percentage of system-on-chip area occupied by embedded memories has led to an increase in the expected number of memory faults. The soft memory repair strategy proposed in this paper employs existing forward error correction at the system level and mitigates the impact of memory faults through permutation of high-sensitivity regions. The effectiveness of the proposed repair technique is evaluated on a multi-megabit de-interleaver static random access memory of an ISDB-T digital baseband orthogonal frequency-division multiplexing receiver in 65-nm CMOS. The proposed technique introduces a single multiplexer delay overhead and a configurable area overhead of $\lceil M/i \rceil$ bits, where M is the number of memory rows and i is an integer from 1 to M, inclusive. The repair strategy achieves a measured 0.15 dB gain improvement at 2×10^{-4} quasi-error-free bit error rate in the presence of stuck-at memory faults for an additive white Gaussian noise channel.**

***Index Terms*— Embedded SRAM memory, fault tolerance, forward error correction (FEC), interleaver, orthogonal frequency-division multiplexing (OFDM) receiver, soft memory repair, system-on-chip (SoC), yield.**

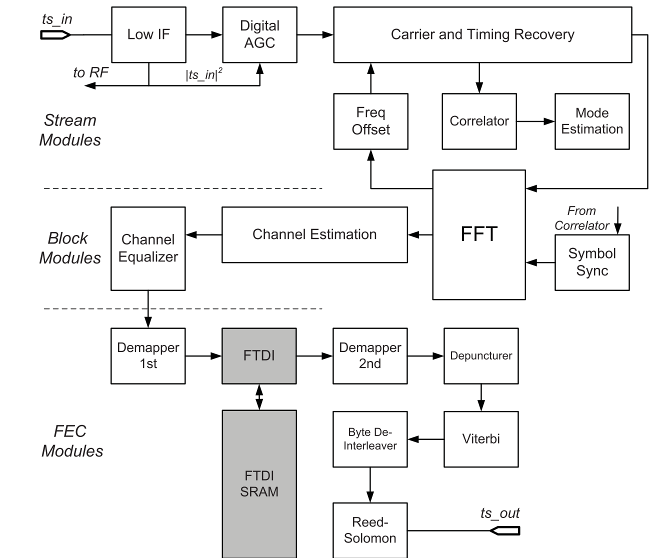

Fig. 1. Generic OFDM digital demodulator architecture with frequency time de-interleaver (FTDI) highlighted.

I. Introduction

The international technology roadmap for semiconductors (ITRS) projects that embedded memories will occupy an increasing percentage of a system-on-chip (SoC) area [1]. As a result, the overall SoC yield is becoming increasingly dependent on memory yield. The high density of integration enabled by diminishing transistor geometries makes embedded memories particularly susceptible to manufacturing faults. Manufacturing process variations also dramatically reduce the reliability and yield of fabricated SoCs. Hence, demand will increase for embedded memories that consume relatively large die areas but are highly adaptable to internal failures. Such designs can help control costs of design verification, manufacturing, and testing [2]–[6].

Repair strategies that utilize redundant resources such as spare rows and columns to repair faulty memory cells [7]–[9] introduce area overhead and contribute to the cost of the SoC. Even though techniques such as divided word and bit lines [10] and redundancy analysis schemes [11] attempt to reduce the overhead, not all memory cells contribute equally to system-level performance. For example, in baseband signal processing, as shown in Fig. 1, a faulty least significant bit (LSB) when compared to a most significant bit (MSB) fault leads to smaller performance degradation as measured by system parameters such as the bit error rate (BER). Similarly, memories that store data prior to filtering and error correction operations exhibit higher fault tolerance due to a higher degree of randomness as measured by entropy. This variation in sensitivity to memory faults can be exploited to minimize the impact of faults, whereby faulty memory blocks with high sensitivity to faults are permuted with functional blocks of low fault sensitivity without resorting to redundant rows and columns. Furthermore, forward error correction (FEC) at the system level can be used to save the area overhead required to implement local error correction at the memory level.

In many statistical signal processing applications, such as digital communications and video processing, a certain number of errors can be tolerated without a noticeable degradation in performance or user experience of the device [12]. As a result, in memory-intensive algorithms considerable area savings can be achieved by mitigating the impact of faults without

Manuscript received November 16, 2011; revised May 5, 2012; accepted June 15, 2012. This work was supported in part by MaxLinear, Inc. and NSERC.

V. Smolyakov and G. Gulak are with the Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: svadim@eecg.toronto.edu; gulak@eecg.toronto.edu).

T. Gallagher and C. Ling are with MaxLinear, Inc., Carlsbad, CA 92011 USA (e-mail: tgallagher@maxlinear.com; cling@maxlinear.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/TVLSI.2012.2208208

1063–8210/$31.00 © 2012 IEEE

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

employing redundant resources and, instead, remapping faulty memory cells containing high-value content with working memory cells containing low-value content. Thus, a fault sensitivity coefficient can be assigned for each memory cell based on a system performance metric such as the BER.

The proposed memory repair strategy eliminates redundant rows and columns in favor of FEC and improves decoding performance in the presence of memory faults by permuting the data so as to minimize the impact of memory faults on system performance as measured by the BER.

II. System Overview

Orthogonal frequency-division multiplexing (OFDM) multicarrier transmission schemes find wide application in wireline as well as wireless standards. The design differences across OFDM receivers supporting different standards can be abstracted and grouped into stream, block, and FEC modules. The stream modules perform synchronization and mode estimation functions. The block modules compute the fast Fourier transform (FFT) and carry out channel estimation and equalization. The FEC modules perform de-interleaving and FEC operations, as illustrated by the soft-output Viterbi and Reed–Solomon decoders.

The area occupied by embedded memory in future OFDM receivers is expected to rise, as well as the fault density. Thus, to address the problem of the increasing number of manufacturing faults, a generic model of an embedded-memory OFDM receiver is presented in Fig. 1.

Without loss of generality, the proposed memory repair strategy is illustrated on an ISDB-T OFDM receiver, and more specifically in the frequency-time de-interleaver (FTDI) because of its large memory requirements as described in the ISDB-T standard [13] and highlighted in Fig. 1. The SRAM-based FTDI occupies more than half of the SoC core area. Thus, it is the single largest area contributor. In addition, due to the high density of embedded SRAM, the probability of SRAM errors per unit area caused by manufacturing faults is several times higher than standard cell digital logic. Thus, area-efficient memory repair strategies must be developed to address the higher probability of SRAM faults.

A. Frequency-Time De-Interleaver

A block diagram of the frequency and time convolutional de-interleaver is shown in Fig. 2. An interleaver changes the order of symbols before transmission to convert long burst errors into shorter bursts or random errors that can be more easily corrected by the error correction logic [14], [15]. Interleavers are characterized by an encoding delay and storage capacity and can take on a convolutional or a block form.

A block interleaver of degree m formats the input symbol vector of length $m \times n$ into a rectangular array of m rows and n columns such that a consecutive pair of symbols at the input appears m symbols apart at the output. The rectangular array is filled row by row and the interleaver output is read out column by column. As a result, an $(n,\ k)$ block code that can handle burst errors of length $b < \lfloor (1/2)(n-k) \rfloor$ when combined with

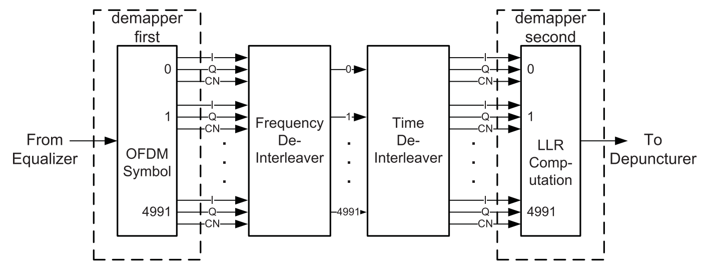

Fig. 2. FTDI block diagram.

TABLE I
Interleaver Characteristics

Type	Storage capacity	Delay
Block	$m \times n$	$m \times (n-1)$
Convolutional	$(m/2) \times (m-1) \times d$	$m \times (m-1) \times d$
Helical	$m \times n$	$m \times n\lceil (m+1)/n \rceil$

an interleaver of degree m creates an interleaved (mn, mk) block code that can handle bursts of length $m \times b$ [16].

A convolutional interleaver of degree m consists of m shift registers with the ith register having a storage capacity of $(i-1) \times d$, for a fixed positive integer d and $i = 1,\ 2,\ 3, \ldots, m$. Each new input symbol is written to a new shift register, while the oldest symbol in the shift register is shifted to the output. Convolutional interleavers reduce the required storage space to approximately half of block interleavers but require a higher degree of clock synchronization. The synchronization period can be reduced with the use of helical interleavers [17].

Table I summarizes the delays and storage capacities for the three types of interleavers.

III. Fault-Tolerant Strategy

In order to develop an efficient fault-tolerant strategy for embedded-memory baseband signal processing systems, it is important to understand the nature of memory faults and to quantify their effect on yield.

A. Yield Model

Yield can be defined as the probability of having zero faults on a chip. Yield can be divided into two classes: gross yield and random fault yield [18]. Gross yield refers to global defects such as incorrect process parameters that can cause large parts of a wafer to have nonfunctional chips. For an m step process, the gross yield can be modeled as

$$Y_{\text{gross}} = \prod_{i=1}^{m} Y_{0i} \tag{1}$$

where $\{Y_{0i} \in \mathbb{R} \mid Y_{0i} \in [0,\ 1]\}$ represents the impact of gross defects in the process step i on the gross yield Y_{gross}.

Random fault yield is based on statistical models of random factors that affect chip yield such as gate oxide pinholes, particle contamination, overlay faults, process-induced shorts and opens, layer thickness, and critical dimension variations. Random faults can be modeled in terms of the average number

TABLE II
MBIST ALGORITHMS [19], [20]

Test name	$O(N)$	Description	Faults covered
MATS	4N	$\{\Updownarrow (w0, r0, w1, r1)\}$;	SAF, SOF
MATS+	5N	$\{\Updownarrow (w0); \Uparrow (r0, w1); \Downarrow (r1, w0)\}$	SAF, DRF
MATS++	6N	$\{\Updownarrow (w0); \Uparrow (r0, w1); \Downarrow (r1, w0, r0)\}$	SAF, SOF, DRF, TF
March C−	10N	$\{\Updownarrow (w0); \Uparrow (r0, w1); \Uparrow (r1, w0); \Downarrow (r0, w1); \Downarrow (r1, w0); \Updownarrow (r0)\}$	SAF, DRF, TF
March B	17N	$\{\Updownarrow (w0); \Uparrow (r0, w1, r1, w0, r1); \Uparrow (r1, w0, w1); \Downarrow (r1, w0, w1, w0); \Downarrow (r0, w1, w0)\}$	SAF, DRF, TF

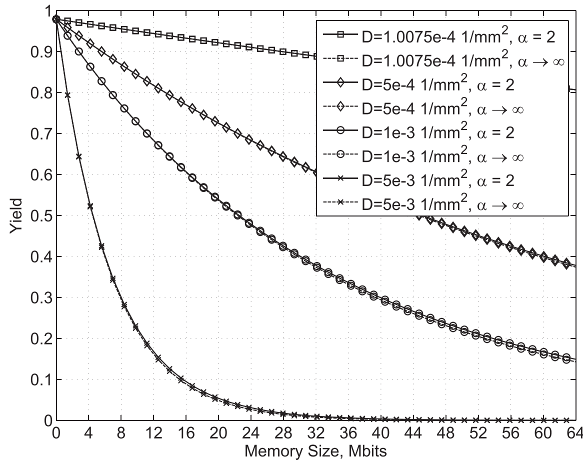

Fig. 3. SoC yield as a function of memory size and fault density based on the negative binomial distribution.

of faults λ_j of type j expressed as a fault density D_j over a critical area A_j, i.e., $\lambda_j = D_j \times A_j$. The statistical distribution of faults on a chip in process step i can be approximated by a Poisson probability distribution $p_{X_i}(k)$ with $E[X_i] = \lambda_i$ [18]. In order to account for chip-to-chip variation of λ_i, a mixed Poisson distribution $Y_{\text{rnd}} \sim \text{Pois}(\Lambda)$ can be used with gamma distribution as a compounder or a mixing function $\Lambda \sim \Gamma(\alpha,\ \lambda/\alpha)$ [21]. The result is a generalized negative binomial distribution given by

$$p_{X_i}(k) = \frac{\Gamma(\alpha_i + k)}{k!\Gamma(\alpha_i)} \frac{(\lambda_i/\alpha_i)^k}{(1+\lambda_i/\alpha_i)^{k+\alpha_i}} \tag{2}$$

with $E[X_i] = \lambda_i$ and $VAR[X_i] = \lambda_i(1+\lambda_i/\alpha_i)$, where the variation in λ_i is modeled by a clustering parameter α_i. Assuming that in each process step i, random faults are independent and identically distributed (iid), Y_{rnd} can be expressed as

$$Y_{\text{rnd}} = \prod_{i=1}^{m} p_{X_i}(0) = \prod_{i=1}^{m} \left(1 + \frac{\lambda_i}{\alpha_i}\right)^{-\alpha_i}. \tag{3}$$

Combining (1) (the gross yield) and (3) (the random yield), the overall SoC yield is

$$Y = \prod_{i=1}^{m} Y_{0i} \prod_{i=1}^{m} \left(1 + \frac{\lambda_i}{\alpha_i}\right)^{-\alpha_i}. \tag{4}$$

Fig. 3 shows a yield versus memory size plot based on (4) with $Y_{0i} = 0.999\ \forall i$, $D_j = 1.0075 \times 10^{-4}\ \text{mm}^{-2}\ \forall j$, $\alpha_i = \{2, \infty\}\ \forall i$, and $m = 21$ process steps. The yield model can be augmented to include fault distributions for any sub-area of the chip as well as the correlation of faults between the sub-areas [22].

B. Fault Model

An embedded memory consists of three main functional blocks: the memory array, the address decoder, and the read and write circuits. The impact of memory faults is different for each functional block. However, faults in the address decoder and the read and write circuits can be modeled equivalently as the corresponding single or multibit faults in the memory array. The memory array faults can be grouped into one of the following categories [23].

1) *Stuck at Faults (SAFs)*: A memory cell value is stuck-at-zero (s-a-0) or stuck-at-one (s-a-1) and the contents of the cell cannot be altered.
2) *Stuck Open Faults (SOFs)*: A memory cell is stuck open and the contents of the cell cannot be accessed.
3) *Data Retention Faults (DRFs)*: A memory cell fails to retain its value after a certain period of time.
4) *Transition Faults (TFs)*: A memory cell fails in at least one $0 \to 1$ or $1 \to 0$ transitions.
5) *Coupling Faults (CFs)*: A state, an operation, or a transition because of a write to one memory cell (coupling cell) affecting the value of another memory cell (coupled cell).

SAFs account for more than 50% of memory array faults [23] and therefore can be used as a first-order approximation to a failure mechanism in a faulty memory array. A fault map showing the location of memory faults can be obtained via a diagnostic memory built-in self-test (MBIST) [24], [25] march tests, in which the address pointer marches through the memory address space writing ($w0$, $w1$) and reading ($r0$, $r1$) bit patterns, and comparing the read-out data with the expected result. Table II summarizes several important march algorithms [19], [20]. For example, MATS+ is defined as $\{\Updownarrow (w0); \Uparrow (r0, w1); \Downarrow (r1, w0)\}$, where $\Updownarrow$, $\Uparrow$, and $\Downarrow$ indicate any, up, and down address order directions, respectively. A memory cell is treated as faulty if a mismatch between expected and received data occurs during an MBIST march test. Thus, the types of faults that can be repaired by the proposed technique are the types of faults that can be detected during the execution of MBIST march tests. Furthermore, fault-tolerant memory repair

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Algorithm 1 Memory Repair Algorithm (Mode, M, N, I)

```
1: if (Mode = MBIST) then
2:   for i = 0 to I − 1 do
3:     row_address_fault[i][M-1:0] ⇐ 0;
4:     if (cur_err_out = 1) then
5:       error_word[i] ⇐ error_register[i][N-1:0];
6:       MSB_region[i] ⇐ error_word[i][sensitivity ≥ threshold];
7:       LSB_region[i] ⇐ error_word[i][sensitivity < threshold];
8:       if (|MSB_region[i] = 1 and |LSB_region[i] = 0) then
9:         row_address_fault[i][(cur_row_address[i])] ⇐ 1;
10:      end if
11:    end if
12:  end for
13: else
14:  while (Mode = Functional) do
15:    if (row_address_fault[i][(cur_row_address[i])] = 1) then
16:      permute (MSB_region, LSB_region);
17:    end if
18:  end while
19: end if
20: return row_address_fault[i]
```

techniques can be classified into hard, soft, combinational, and cumulative repair strategies [1] based on how the repair information is acquired and retrieved.

C. Proposed Repair Strategy

The proposed repair strategy saves implementation costs by eliminating redundant rows and columns or local error correction in favor of FEC and improving decoding performance in the presence of memory faults by permuting the data so as to minimize the impact of memory faults on system performance as measured by the BER. The proposed repair technique assigns a fault sensitivity coefficient for each memory cell based on the impact of cell fault on a system performance metric such as the BER. Thus, each addressable word in the memory array is divided into fields or blocks of high and low sensitivity to memory cell faults.

To minimize the impact of memory faults on system performance, the data block is permuted such that bits with higher fault sensitivity coefficients are assigned fault-free memory locations, while bits with lower fault sensitivity coefficients are assigned faulty memory locations. A sensitivity coefficient ζ is assigned for each bit in a memory word as a difference in BER caused by a SAF compared to the fault-free memory cell, normalized to 1

$$\zeta = \frac{1}{C}(\mathrm{BER}_{SA} - \mathrm{BER}_{FF}) \quad (5)$$

where the subscripts SA and FF represent stuck-at and fault-free cases, respectively, and C is a normalization constant.

Fig. 4 shows a segment of the de-interleaver memory used to store soft I and Q data along with the carrier-to-noise (CN)

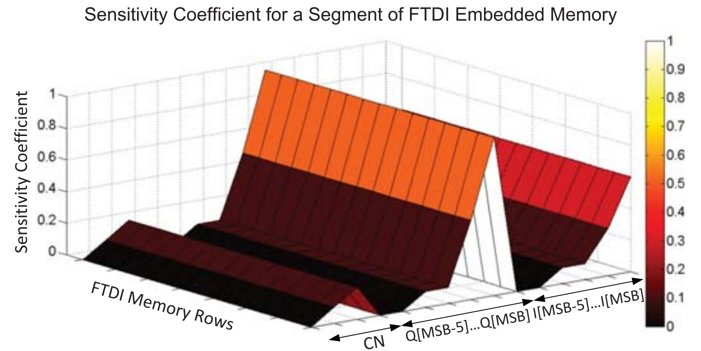

Fig. 4. Fault sensitivity coefficient for a segment of FTDI embedded memory. Memory column segment stores {I[MSB:MSB-5], Q[MSB:MSB-5], CN[MSB:MSB-2]} associated with OFDM symbols stored in memory rows.

ratio. Fig. 4 was obtained on the basis of simulation results described in Section V-A. As expected, the data bits of I, Q, and CN that are closest to the MSB have a higher sensitivity coefficient compared to bits that are farthest away from the MSB.

Thus, the impact of memory faults on system performance can be minimized if MSB data is permuted with LSB data when the MSB memory region contains faulty memory cells while the LSB region is fault free. The proposed soft memory repair technique without redundant rows and columns is summarized in the memory repair algorithm, which operates on I memory instances of size $M \times N$ in parallel during MBIST and functional modes.

The proposed memory repair algorithm initializes the row address fault register in test mode (steps 1–12) by setting a bit corresponding to the location (but not type) of fault in the faulty row address to a "1," and checks the row address fault register in functional mode (steps 13–20) on every memory read and write operation to determine when to activate the permutation logic. In the MBIST mode, the bit error location is captured by reading the error register (5). Next the high- and low-sensitivity regions (determined by the sensitivity coefficient threshold) are examined for the presence of errors via the reduction OR operation (6)–(8), and the row address is labeled faulty if errors are found in the high-sensitivity (MSB) region, while the low-sensitivity (LSB) region is error free (9). In the functional mode, the row address fault register is accessed on every memory operation (15) and, if the current row address is marked faulty, the MSB and LSB regions are permuted (16). Thus, the algorithm provides memory repair without redundant rows and columns in which data that are sensitive to error are stored in error-free memory locations while data that are less sensitive to error are assigned to faulty memory locations.

IV. VLSI Architecture

The proposed repair strategy interfaces with MBIST memory wrappers and integrates with a design-for-test (DFT) on-chip infrastructure.

A. DFT Architecture

Fig. 5 shows the SoC-level DFT architecture. It consists of STAR memory system (SMS) modules, JPC/SFP server, eFuse

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

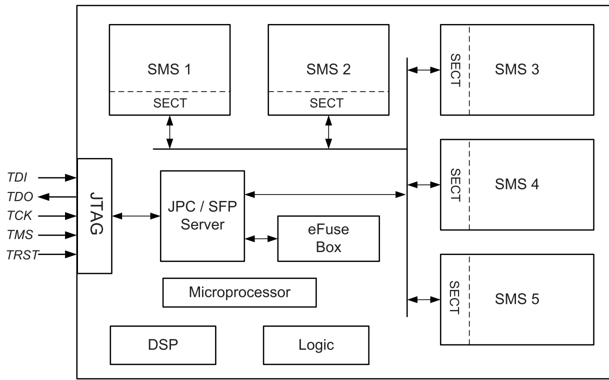

Fig. 5. SoC-level DFT architecture with JTAG IEEE 1149.1 interface [26].

Box, and 1149.1 joint test action group (JTAG) and P1500 standard for embedded core test (SECT) interfaces. The SMS modules contain embedded memory wrappers controlled by a self test and repair (STAR) processor [26]. The JPC/SFP server interfaces IEEE 1149.1 JTAG with IEEE P1500 SECT and provides a connection to the one-time-programmable eFuse Box used to store hard repair information.

B. Repair Architecture

Fig. 6 shows the architecture of the proposed soft memory repair technique. The proposed technique interfaces with MBIST via data and test address bus lines as well as an error signal (cur_err_out) indicating a mismatch during an MBIST march test. In the MBIST mode, the error capture and repair enable logic is used to: 1) capture externally the serial output of the error register; 2) examine its contents for the location of faults in both high- and low-sensitivity regions; and 3) set the corresponding bit of the row address fault register if the higher sensitivity region has at least one fault while the lower sensitivity region is fault free. In the functional mode, the row address fault register is accessed on every read and write operation and, if the current row address is labeled faulty, the regions of high and low sensitivity are permuted by the repair interleave (ITL) logic and output through the 2-to-1 MUX controlled by repair enable signals.

The FTDI memory is organized internally into 1K rows. Therefore, the maximum size of the row address fault register is 1024. However, to reduce area overhead, a single bit in the row address fault register can be used to track multiple rows. Thus, the size of the row address fault register can be reduced by i, where i is an integer between 1 and M, equal to the number of memory rows assigned to a single bit of the row address fault register.

The proposed technique introduces a single multiplexer delay overhead since the only additional data path delay is due to the 2-to-1 MUX during write and read operations, while the ITL logic performs a negligible delay permutation operation. The proposed technique introduces a configurable area overhead of $\lceil M/i \rceil$ bits, where M is the number of memory rows and i is an integer from 1 to M, inclusive. Thus, for the frequency time de-interleaver SRAM memory in 65-nm CMOS, the proposed memory repair algorithm can be configured to introduce 1.7% of area overhead, when $i = 4$ and $M = 1024$, due to the external-to-memory row address fault register consisting of $M/i = 1024/4 = 256$ flip-flops. The value of $i = 4$ was selected to match the area overhead of the proposed technique with [26] for the purpose of performance comparison. The implementation costs of the proposed repair technique based on the worst case PVT synthesis in 65-nm CMOS are presented in Table III. The repair overhead is summarized under Δ_i columns for $i = 1,\ 2$, and 4, where i is the number of memory rows assigned to a single bit of the row address fault register. The area overhead of the proposed technique was computed by comparing the synthesis area estimates of MBIST memory with and without the proposed repair logic.

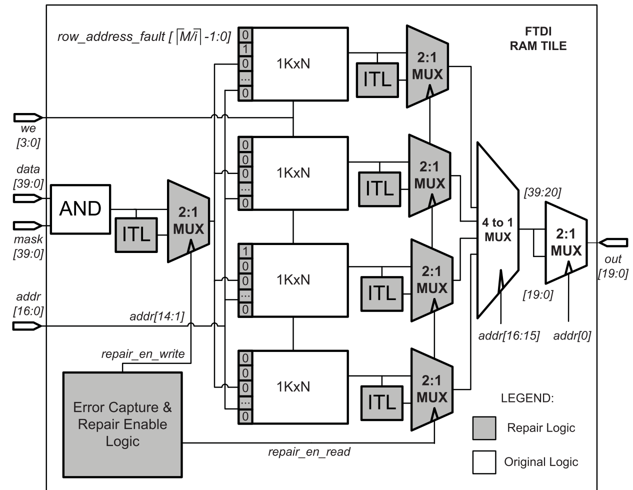

Fig. 6. Proposed memory repair architecture.

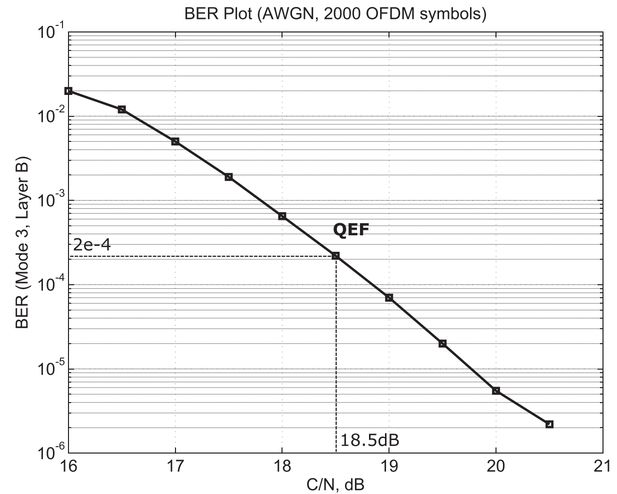

Fig. 7. Simulated BER plot (mode 3, layer B: 64-QAM, $R = 3/4$, $N_R = 1$, AWGN, 2000 OFDM symbols).

V. RESULTS

A. Simulation Results

Fig. 7 shows the BER plot of the OFDM receiver for an additive white Gaussian noise (AWGN) channel with soft-output Viterbi and Reed–Solomon (204, 188) FEC, when the

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE III
ASIC SYNTHESIS RESULTS OF THE PROPOSED MEMORY REPAIR IN 65-nm CMOS FOR THE WORST CASE PVT: SS, 1.08 V, 125 °C. Δ_i: REPAIR OVERHEAD WHEN THE FAULTY ROW ADDRESS IS RECORDED FOR EVERY i MEMORY ROWS

65 nm CMOS	FTDI SRAM (original)	FTDI SRAM ($i = 1$)	Δ_1	FTDI SRAM ($i = 2$)	Δ_2	FTDI SRAM ($i = 4$)	Δ_4
Clock rate (MHz) (Spec. = 64 MHz)	69.4	69.4	4 ps w.c. slack	69.4	4 ps w.c. slack	69.4	4 ps w.c. slack
% Area overhead	–	–	5.2%	–	2.5%	–	1.7%
Number of std. cells	14 074	144031	129957	78 646	64 572	42 943	28 869
Dynamic (μW)	31 060	39 413	8352	32 845	1785	32 387	1327
Leakage (μW)	887	1055	168	987	100	928	41
Total power (μW) @ 69.4 MHz	31 947	40 468	27%	33 832	5.9%	33 315	4.3%

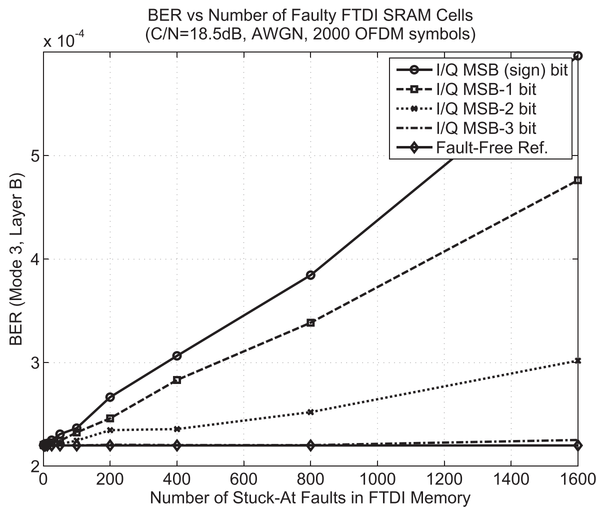

Fig. 8. Simulated BER plot (mode 3, layer B: 64-QAM, $R = 3/4$, $N_R = 1$, AWGN, 2000 OFDM symbols).

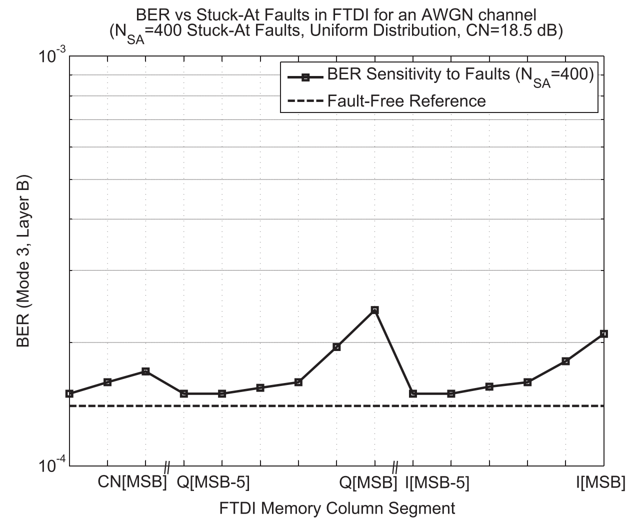

Fig. 9. Simulated BER plot (mode 3, layer B: 64-QAM, $R = 3/4$, $N_R = 1$, AWGN, 2000 OFDM symbols).

de-interleaver memory is fault free. The quasi-error-free (QEF) point is defined as the maximum acceptable BER for which the application or the user does not perceive any degradation in performance. The ISDB-T QEF point for an AWGN channel is 2×10^{-4} at 18.5 dB carrier-to-noise (CN) ratio.

Fig. 8 shows the sensitivity of BER when N_{SA} SAFs, alternating between s-a-0 and s-a-1, are uniformly distributed throughout each group of 190 OFDM symbols, corresponding to the worst case de-interleaver memory delay for layer B [13], for the top four bits of I/Q at the QEF point for an AWGN channel. SAFs were introduced via a function that modified the memory array via bitwise OR operations with a 1 for s-a-1 faults, and bitwise AND operations with a 0 for s-a-0 faults. According to the simulation results in Fig. 8, the MSB (sign bit) shows a higher sensitivity to N_{SA} faults in comparison to MSB-3 bit, which is close to the fault-free reference.

Fig. 9 shows the impact of $N_{SA} = 400$ SAFs on BER for I[MSB:MSB-5], Q[MSB:MSB-5], CN[MSB:MSB-2] memory column segment of the frequency time de-interleaver. The fault sensitivity coefficient ζ in Fig. 4 was computed on the basis of the BER plot in Fig. 9. By setting a sensitivity threshold to 1.6×10^{-4}, or 7% above the fault-free reference, the high-sensitivity (MSB) region consists of the top three bits of I and Q and the MSB of CN: $\{I[\text{MSB}: \text{MSB} - 2], Q[\text{MSB}: \text{MSB} - 2], CN[\text{MSB}]\}$, while the low-sensitivity region of the same width contains $\{I[\text{MSB} - 3: \text{MSB} - 5], Q[\text{MSB} - 3: \text{MSB} - 5], CN[\text{MSB} - 1]\}$. The gain improvement ΔCN_{sim} due to the proposed repair was found by dividing the maximum difference in the MSB BER in Fig. 9 by the slope of the BER plot at the QEF BER $= 2 \times 10^{-4}$ in Fig. 7: $\Delta CN_{\text{sim}} = (\text{BER}_{Q[\text{MSB}-3]} - \text{BER}_{Q[\text{MSB}]})/\text{QEF slope} = (0.00015 - 0.00024)/(-2.6 \times 10^{-4}\ \text{dB}^{-1}) = 0.35$ dB. Thus, if a memory fault is found in the high-sensitivity region and no faults were found in the low-sensitivity region, the permutation of high-sensitivity regions in the case of Fig. 9 results in 0.35 dB gain improvement at 2×10^{-4} BER over memory without repair at the QEF BER for an AWGN channel.

B. Measurement Results

The hardware test setup used to verify the proposed memory repair strategy consists of an ISDB-T signal generator (LG3802), a wireless channel emulator (SR5500), and a custom FPGA platform connected to a PC via an I^2C interface. The FPGA platform consists of two Virtex-5 LX330 FPGAs,

in addition to an RF tuner card used to interface to the channel emulator and an external-to-FPGA SRAM memory chip used to store the de-interleaver data because of its large memory requirements.

A fault mask is used to introduce bursts of alternating s-a-0 and s-a-1 faults of length N_{SA}, distributed throughout every group of 190 OFDM symbols, before the data are written into the functional SRAM chip, acting as a faulty de-interleaver memory. Measurement results were recorded by reading the internal registers of the OFDM receiver via an I^2C interface. Each point on the BER curve is based on the average value of the BER register over a 3-min interval, corresponding to the transmission of a payload of approximately 20 Mb/s $\times$ 180 s $=$ 3.6 Gbits.

Fig. 10 shows the deviation in QEF BER for the high $N_{SA} = 400$ fault case, with and without proposed memory repair for an AWGN channel.

The high-sensitivity region consists of $\{I[\text{MSB: MSB} - 3]$, $Q[\text{MSB: MSB} - 3], CN[\text{MSB: MSB} - 1]\}$, while the low-sensitivity region is defined as $\{I[\text{MSB} - 4: \text{MSB} - 7]$, $Q[\text{MSB} - 4: \text{MSB} - 7], CN[\text{MSB} - 2: \text{MSB} - 3]\}$.

The dashed line shows an increase in measured BER for the de-interleaver memory with N_{SA} faults. The solid line represents measured BER when the proposed memory repair is enabled. As a result of the permutation of MSB and LSB regions, the proposed repair strategy achieves fault sensitivity exhibited by the LSB region for MSB region data whenever the MSB region has faults and the LSB region is fault free.

The gain improvement ΔCN_{meas} due to the proposed repair is calculated by dividing the maximum difference in the MSB BER in Fig. 10 by the slope of the BER plot at the QEF BER $= 2 \times 10^{-4}$ in Fig. 7: $\Delta CN_{\text{meas}} = (\text{BER}_{I[\text{MSB}-4]} - \text{BER}_{I[\text{MSB}]})/QEF\text{slope}) = (0.00012 - 0.00016)/(-2.6 \times 10^{-4}\ \text{dB}^{-1}) = 0.15$ dB. The measured ΔCN_{meas} is 0.2 dB smaller than the simulated ΔCN_{sim}. The 0.2 dB loss is attributed to the RF tuner card, which was not modeled in the simulation.

Fig. 11 shows the deviation in QEF BER for a six-path fading channel in the presence of N_{SA} burst faults distributed throughout every group of 190 OFDM symbols with and without the proposed memory repair for the top four bits of I and Q. The dashed line represents the proposed memory repair and shows a smaller QEF BER degradation in comparison to memory without repair over all Doppler frequencies [10, 20, 30, 40] Hz studied. In the case of the MSB fault and $F_d = 40$ Hz for a TU-6 channel, the proposed repair reduces the BER from 0.00245 to 0.00235 or 4.1% decrease with soft-output Viterbi and Reed–Solomon (204, 188) FEC.

C. Discussion

Table IV compares implementation performance of different SRAM memory repair techniques. The proposed strategy introduces a single multiplexer latency overhead on read and write operations and a configurable area overhead dominated by external-to-memory fault registers of size $\lceil M/i \rceil$ bits, where M is the number of memory rows and i is an integer between 1 and M, inclusive. The proposed repair technique

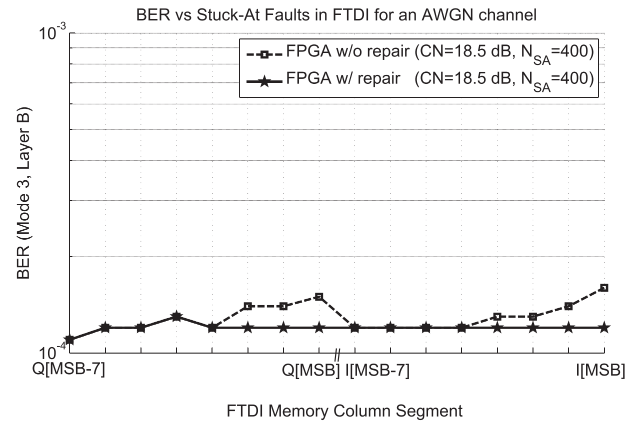

Fig. 10. Measured results (FPGA): BER versus SAFs in FTDI memory segment for an AWGN channel.

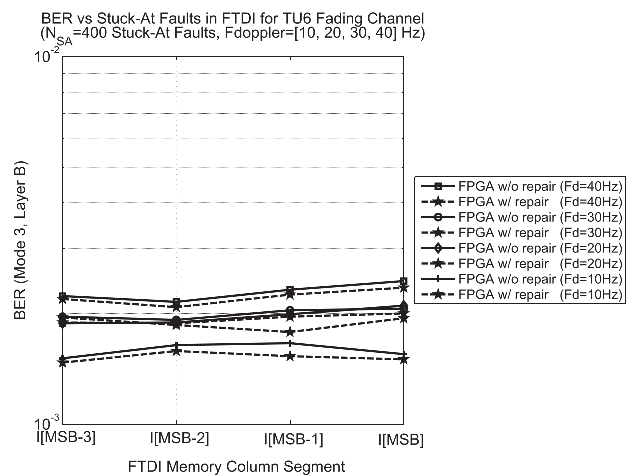

Fig. 11. Measured results (FPGA): BER versus SAFs in FTDI memory segment for a TU-6 fading channel.

is different in the sense that it seeks to minimize the impact of embedded memory faults through permutation of high-sensitivity regions in addition to employing downstream soft-output Viterbi and Reed–Solomon decoders for correcting memory faults rather than using costly redundancy in the form of spare rows and columns or local ECC for memory repair. The repair technique in [26] is a straight-forward memory repair implementation based on column redundancy integrated in the layout of the main memory. The proposed repair technique was configured to introduce a comparable area overhead for the purpose of repair performance comparison. Note that the proposed technique can be configured to save a greater percentage of memory area dedicated to repair. Moreover, the repair performance in [26] with redundancy is limited by the number of spare rows and columns, while the proposed technique is capable of permuting data for all faulty memory rows. The choice of using external-to-memory fault registers for storing the repair information for every memory instance increases the gate count in comparison to [7] and [8]. The large gate count is a result of the row address fault register of size $\lceil M/i \rceil$ bits for each de-interleaver memory instance with $M = 1024$ and $i = 1, 2, 3, \ldots, M$, where i is the number of memory rows assigned to a single bit of the

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE IV
SRAM MEMORY REPAIR PERFORMANCE COMPARISON

Parameter	[7]-2006	[8]-2007	[9]-2010	[26]-2011	This paper-2011*
Technology	180-nm	180-nm	180-nm	65-nm	65-nm
Area overhead	6.5%	2.8%	2.3%	1.7%	1.7%
Gate count [kGE]	6.3	8.3	N/A	N/A	38.1
Redundant rows	4	3	6	0	0
Redundant cols.	2	3	6	4	0
Error correction	No	No	No	No	Yes
Clk frequency [MHz]	N/A	N/A	N/A	69.4	69.4
Repair strategy	Soft	Soft	Soft	Hard	Soft

*Based on $i = 4$ in Table III.

row address fault register. The value of i can be adjusted to reflect the expected number of faults λ for a given technology process. For example, the value of i set to $M/4$ introduces an area overhead of only 4 flip-flops per memory instance. Thus, by tuning the parameter i, one can trade off area overhead with the effectiveness of the proposed repair technique. Alternatively, the fault register can be implemented as a block of memory of size $\lceil M/i \rceil$ for each instance or as a separate memory. The repair technique in [9] uses a global block-level repair approach for eliminating clustering faults to minimize the number of required spares. In comparison, the proposed technique reduces the impact of clustering faults through local block-level permutation of programmable sensitivity regions. The timing penalty of a single multiplexer delay on read and write operation is comparable to [7]; however, no write buffer is required since the data is permuted via combinational logic before it is written to or read from the memory. The effectiveness of the proposed repair technique was evaluated on a system-level performance metric such as the BER for the frequency–time de-interleaver memory. The impact of memory faults on the BER can be found for other baseband subsystems such as LDPC [4], turbo [6], and Viterbi [27] decoders. A large number of SAFs was selected to account for memory faults not modeled in the simulation and to test the system under high fault conditions. While SAFs were considered, the repair technique is not limited to hard memory faults and can also be applied to soft faults induced by reducing memory supply voltage in order to lower the power consumption [27]. In addition, the repair permutation block can be hard-wired to reduce implementation complexity or programmable, e.g., a permutation network. For example, a Benes permutation network can be used to adapt the permutation of high-sensitivity regions to a variety of data formats and to account for a potential difference between the logical address and the corresponding physical memory locations that may arise due to memory layout constraints [28]. The proposed repair technique is integrated with a commercial BIST infrastructure, similar to [8]; however, it is generic enough to be used with a variety of memory BIST hardware.

The 0.15 dB gain in Fig. 10 represents the measured improvement of the proposed repair technique compared to memory without repair and includes RF card losses. Memories with larger word length and therefore larger separation between MSB and LSB are expected to have higher performance gains.

The limitations of the proposed technique are that it requires an existing MBIST infrastructure for interfacing with the proposed repair logic, an existing FEC mechanism for improved performance, and that it seeks to mitigate the impact of memory faults on the BER (through permutation of fault sensitivity regions and FEC) rather than eliminating the faults via limited and costly repair rows and columns. In addition, system-level simulations are required to determine the boundaries of sensitivity regions for each of the embedded memories within each baseband subsystem of the SoC. The advantages of the proposed method are memory area savings achieved by eliminating redundant rows and columns, a single multiplexer delay overhead, configurable area overhead, a simple interface with an existing MBIST infrastructure, and programmable sensitivity regions.

VI. CONCLUSION

A soft memory repair strategy for baseband signal processing systems without redundant spare rows and columns has been proposed. The proposed repair strategy saves implementation costs by eliminating redundancy or local error correction in favor of FEC at the system level and improves decoding performance in the presence of memory faults by permuting the data so as to minimize the impact of memory faults on the BER. The effectiveness of the proposed repair technique is demonstrated on a multi-megabit de-interleaver SRAM memory of an ISDB-T digital baseband OFDM receiver in 65-nm CMOS. The proposed technique introduces a single multiplexer delay overhead and a configurable area overhead of $\lceil M/i \rceil$ bits, where M is the number of memory rows and i is an integer from 1 to M, inclusive. The proposed repair strategy achieves a measured 0.15 dB gain improvement at 2×10^{-4} QEF BER in the presence of memory errors for an AWGN channel.

ACKNOWLEDGMENT

The authors would like to thank MaxLinear, Inc., Carlsbad, CA, and NSERC for supporting this work.

REFERENCES

[1] Y. Zorian, “Embedded-memory test and repair: Infrastructure IP for SoC yield,” in *Proc. Int. Test Conf.*, 2002, pp. 340–348.
[2] *Design Report*, ITRS, Tsukuba, Japan, 2009, pp. 1–42.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[3] J.-F. Li, T.-W. Tseng, and C.-S. Hou, “Reliability-enhancement and self-repair schemes for SRAMs with static and dynamic faults,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 9, pp. 1361–1366, Sep. 2010.

[4] M. May, M. Alles, and N. Wehn, “A case study in reliability-aware design: A resilient LDPC code decoder,” in *Proc. Design, Autom. Test Eur.*, Mar. 2008, pp. 456–461.

[5] C. Brehm, M. May, C. Grimmler, and N. Wehn, “A case study on error resilient architectures for wireless communication,” in *Proc. Arch. Comput. Syst.*, 2012, pp. 13–24.

[6] A. M. Eltawil and F. J. Kurdahi, “System redundancy; a means of improving process variation yield degradation in memory arrays,” in *Proc. Int. Symp. VLSI Des., Autom. Test*, Apr. 2006, pp. 1–4.

[7] L.-M. Denq, T.-C. Wang, and C.-W. Wu, “An enhanced SRAM BISR design with reduced timing penalty,” in *Proc. 15th Asian Test Symp.*, 2006, pp. 25–30.

[8] C.-D. Huang, J.-F. Li, and T.-W. Tseng, “ProTaR: An infrastructure IP for repairing RAMs in system-on-chips,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 10, pp. 1135–1143, Oct. 2007.

[9] S.-K. Lu, C.-L. Yang, Y.-C. Hsiao, and C.-W. Wu, “Efficient BISR techniques for embedded memories considering cluster faults,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 2, pp. 184–193, Feb. 2010.

[10] S.-K. Lu and C.-H. Hsu, “Fault tolerance techniques for high capacity RAM,” *IEEE Trans. Rel.*, vol. 55, no. 2, pp. 293–306, Jun. 2006.

[11] S. K. Thakur, R. A. Parekhji, and A. N. Chandorkar, “On-chip test and repair of memories for static and dynamic faults,” in *Proc. Int. Test Conf.*, 2006, pp. 1–10.

[12] F. J. Kurdahi, A. M. Eltawil, P. Young-Hwan, R. N. Kanj, and S. R. Nassif, “System-level SRAM yield enhancement,” in *Proc. Int. Symp. Qual. Electron. Des.*, Mar. 2006, pp. 178–184.

[13] *Transmission System for Digital Terrestrial Television Broadcasting*, Standard STD-B31, Nov. 2005.

[14] J. L. Ramsey, “Realization of optimum interleavers,” *IEEE Trans. Inf. Theory*, vol. 16, no. 3, pp. 338–345, May 1970.

[15] D. Forney, “Burst-correcting codes for the classic bursty channel,” *IEEE Trans. Commun. Technol.*, vol. 19, no. 5, pp. 772–781, Oct. 1971.

[16] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. New York: McGraw-Hill, 2008.

[17] E. R. Berlekamp and P. Tong, “Interleavers for digital communications,” U.S. Patent 4 559 625, Dec. 17, 1985.

[18] C. Stapper, F. Armstrong, and K. Saji, “Integrated circuit yield statistics,” *Proc. IEEE*, vol. 71, no. 4, pp. 453–470, Apr. 1983.

[19] A. van de Goor, C. Jung, S. Hamdioui, and G. Gaydadjiev, “Low-cost, customized and flexible SRAM MBIST engine,” in *Proc. Int. Symp. Des. Diag. Electron. Circuits Syst.*, 2010, pp. 382–387.

[20] S. M. Al-Harbi and S. K. Gupta, “An efficient methodology for generating optimal and uniform march tests,” in *Proc. VLSI Test Symp.*, 2001, pp. 231–237.

[21] M. Ottavi, L. Schiano, X. Wang, Y.-B. Kim, F. J. Meyer, and F. Lombardi, “Evaluating the yield of repairable SRAMs for ATE,” *IEEE Trans. Instrum. Meas.*, vol. 55, no. 5, pp. 1704–1712, Oct. 2006.

[22] I. Koren and Z. Koren, “Defect tolerance in VLSI circuits: Techniques and yield analysis,” *Proc. IEEE*, vol. 86, no. 9, pp. 1819–1836, Sep. 1998.

[23] R. Dekker, F. Beenker, and L. Thijssen, “Fault modeling and test algorithm development for static random access memories,” in *Proc. IEEE Int. Test Conf.*, Sep. 1988, pp. 343–352.

[24] V. D. Agrawal, C. R. Kime, and K. Saluja, “A tutorial on built-in self-test. I. Principles,” *IEEE Des. Test Comput.*, vol. 10, no. 1, pp. 73–82, Mar. 1993.

[25] V. D. Agrawal, C. R. Kime, and K. Saluja, “A tutorial on built-in self-test. 2. Applications,” *IEEE Des. Test Comput.*, vol. 10, no. 2, pp. 69–77, Jun. 1993.

[26] *STAR Memory System, Virage Logic Product Manual*, Synopsys, Mountain View, CA 2011.

[27] A. M. Hussein, M. S. Khairy, A. Khajeh, K. Amiri, A. M. Eltawil, and F. J. Kurdahi, “A combined channel and hardware noise resilient Viterbi decoder,” in *Proc. Asilomar Conf. Signals, Syst. Comput.*, Nov. 2010, pp. 395–399.

[28] A. van de Goor and I. Schanstra, “Address and data scrambling: Causes and impact on memory tests,” in *Proc. 1st IEEE Int. Workshop Electron. Des., Test, Appl.*, Jan. 2002, pp. 128–136.

Vadim Smolyakov (S’05) received the B.A.Sc. degree (Hons.) in engineering science specializing in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 2009, where he is currently pursuing the M.A.Sc. degree in electrical and computer engineering.

He held numerous positions as a Research Assistant with the Department of Electrical and Computer Engineering, University of Toronto. He was a Communication Systems Engineer working on a cross-level optimization of a system-on-chip OFDM receiver with MaxLinear, Inc., Carlsbad, CA, from January 2011 to July 2011. His current research interests include signal processing algorithms and VLSI architectures for digital communication, multimedia, and biomedical applications.

Mr. Smolyakov was a recipient of the Natural Sciences and Engineering Research Council of Canada (NSERC) graduate scholarship.

Glenn Gulak (S’82–M’83–SM’96) received the Ph.D. degree from the University of Manitoba, Winnipeg, MB, Canada, in 1984.

He was a Research Associate with the Information Systems Laboratory and the Computer Systems Laboratory, Stanford University, Stanford, CA, from 1985 to 1988. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. His current research interests include memory design, circuits, algorithms, and VLSI architectures for digital communication.

Dr. Gulak was a recipient of the Postgraduate Scholarship from the Natural Sciences and Engineering Research Council of Canada and several teaching awards for undergraduate courses taught in the Department of Computer Science and the Department of Electrical and Computer Engineering, University of Toronto, where he was a recipient of the L. Lau Chair. He was the Technical Program Chair of the International Solid State Circuits Conference in 2001. He is a registered Professional Engineer in the province of Ontario.

Timothy Gallagher (M’05) received the B.S. degree in electrical engineering and computer science from the University of Colorado, Boulder, and the M.S. degree in electrical engineering, specializing in signal processing, from the University of Southern California, Los Angeles, in 1982 and 1988, respectively.

He is currently a Vice President of communication systems with MaxLinear, Inc., Carlsbad, CA. His current research interests include efficient implementation of signal processing algorithms and digital communication.

Curtis Ling (SM’02) received the B.S. degree in electrical engineering from the California Institute of Technology, Pasadena, and the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor.

He is a Co-Founder of MaxLinear, Inc., Carlsbad, CA, where he has been the Chief Technical Officer since April 2006, was the Chief Financial Officer from 2004 to 2006, and was a Consultant from 2003 to 2004. He was a Principal Engineer with Silicon Wave, Inc., from 1999 to 2003. He was a Professor with the Hong Kong University of Science and Technology, Hong Kong, from 1993 to 1999.