Circuit-Level Timing-Error Acceptance for Design of Energy-Efficient DCT/IDCT-based Systems

Ku He, *Member, IEEE*, Andreas Gerstlauer, *Senior Member, IEEE*, and Michael Orshansky

***Abstract*—An intrinsic notion of quality floors present in typical digital signal processing (DSP) circuits can be used to strategically accept some run-time errors in exchange for a reduction in energy consumption. Conventional VLSI design strategies do not exploit this degree of error tolerance and aim to guarantee timing correctness, thereby sacrificing energy efficiency. In this paper, we propose techniques for Timing ERRor Acceptance (TERRA) to improve the quality-energy tradeoff in image and video processing systems under scaled V_{DD}. The basic philosophy is to prevent signal quality from severe degradation, on average, by using data statistics. The introduced innovations include techniques for carefully controlling possible errors and exploiting the specifics of error patterns for low-cost post-processing to minimize quality degradation.**

We demonstrate the effectiveness of the proposed techniques on a 2D-IDCT and a 2D-DCT design. The designs were synthesized using a 45nm standard cell library, with energy and delay evaluated using NanoSim and VCS. Experiments show that direct application of controlled error-acceptance techniques allows up to 59% and 71% energy savings by permitting fewer than 1dB peak signal-to-noise ratio (PSNR) decrease for the 2D IDCT and DCT designs, respectively. The resulting PSNR remains above 30dB, which is a commonly accepted value for lossy image and video compression. Achieving such energy savings by direct V_{DD} scaling without the proposed transformations results in a 12dB PSNR loss. The area overhead for the needed control logic is about 4.8% of the original design. To further minimize quality degradation caused by accepted errors in the IDCT, we introduce post-filtering on the output image. The significant improvement of the perceived image quality allows further voltage scaling leading to overall energy savings of 70% for the 2D-IDCT, while costing an additional 1.1% in area.

***Index Terms*—Error tolerant computing, low energy design**

I. Introduction

The gap between the limited battery life and the need to support more complex functionality of embedded systems is growing. Mitigating this gap requires continued advances in low energy design. In this work, we propose to exploit error-tolerance of certain signal processing circuits to reduce their energy consumption. Our strategy focuses on circuit-level Timing ERRor Acceptance (TERRA) as a way to reduce energy. In a conventional design methodology, driven by static timing analysis, timing correctness of all operations is guaranteed by construction. The design methodology guarantees that every circuit path regardless of its likelihood of excitation must meet timing. Traditional design strategies do not consider the possibility of accepting timing errors. When V_{DD} is scaled even slightly, large timing errors occur and rapidly degrade the output signal quality. This rapid quality loss under voltage scaling significantly reduces the potential for energy reduction. In this paper, we show how the above quality-energy tradeoff can be dramatically improved.

The proposed TERRA strategy is based on a statistical treatment of errors: while we give up on guaranteeing the worst-case timing, we have to satisfy timing requirements on average to keep global signal quality from severe degradation. We advance architecture-level techniques that significantly reduce algorithm quality loss under V_{DD} scaling, as compared to direct V_{DD} reduction. This leads to a superior quality-energy tradeoff profile. Fundamentally, this is enabled by (i) reducing the occurrence of early timing errors with large impact on quality, (ii) using control and data flow analysis to disallow errors that are spread and get amplified as they propagate through the algorithm, and (iii) applying post-processing techniques to reduce localized large magnitude errors that significantly degrade perceived image quality.

We specifically focus on the behavior of timing errors in addition as a fundamental building block of most signal, image and video processing algorithms. Simple analysis shows that the onset and magnitude of timing errors depends on the values of operands. Targeting the earliest and worst errors, we present four quality-energy (Q-E) optimizations at the operation, block, algorithm and system levels. Techniques are introduced and demonstrated on the designs of an Inverse Discrete Cosine Transform (IDCT) and a Discrete Cosine Transform (DCT) as widely used image and video processing kernels. Note that depending on knowledge about data statistics, techniques can be applied at design or at run time. For the design chosen in this paper, however, we limit discussions to static operation- and algorithm-level and dynamic block- and system-level optimizations.

The rest of the paper is organized as follows: after a discussion of related work in Section II, Sections III and IV discuss the techniques for timing error control and post-processing, respectively. Section V then shows experimental results, while Section VI concludes and summarizes the paper.

II. Related Work

Several efforts in the past have explored the possibility of trading quality in DSP systems for lower energy. In [2], [3], [4], energy is reduced by discarding algorithm steps or iterations that contribute less to the final quality. Various

Manuscript received July 09, 2012; revised October 24, 2012. An earlier version of this work appeared in [1].

K. He is with Cirrus Logic, Inc. Austin, Texas, U.S.A. (e-mail: kuhe@utexas.edu).

A. Gerstlauer and M. Orshansky are with the Department of Electrical and Computer Engineering, The University of Texas at Austin, U.S.A. (e-mail: gerstl@ece.utexas.edu; orshansky@mail.utexas.edu).

approaches employ adaptive precision [5] or approximations [6] in the basic arithmetic units to save energy. In [7], [8], energy reduction is enabled by using lower voltage on a main computing block and employing a simpler error-correcting block that runs at a higher voltage and is thus, error-free, to improve the results impacted by timing errors of the main block. In [9], a low-power DCT core is implemented by identifying and skipping the unnecessary computations. In [10], power is reduced by applying aggressive voltage scaling to the memory of a multimedia system, and then filtering out the resulting memory faults. The most similar approach to ours is described in [11], [12], [13]. In this work, combinational logic blocks are restructured to enable utilization of intermediate results, which are arranged such that the more important ones, from the quality point of view, are obtained first.

An important distinction between prior work and our TERRA strategy is that in other work, the results produced by blocks subject to timing errors are not directly accepted. From the point of view of gate-level design, such techniques still guarantee timing correctness of all operations. In [7], [8], an estimated value of the result is used in downstream computation in case of timing errors. In [11], [12], [13], computation is terminated early and intermediate results impacted by timing errors are ignored entirely. By contrast, our strategy allows using the erroneous results directly, providing, of course, that the magnitude of error is carefully controlled. As a result, we are able to achieve large energy savings in the low range of quality loss. This is similar to the approach in [14], in which path delay shaping is introduced to reduce timing errors in arithmetic operations and an error-detecting control loop is used to monitor and regulate large error rates. Our approach is orthogonal and instead works with unmodified basic components, where errors are controlled through architecture-level design without the area and delay overhead (of up to 20% in [14]) of a complex error detection and control circuit.

The available data from literature suggests that our design is effective. The energy savings are higher than in earlier work: for example, savings are 20% in [14], 55% in [9], 40% in [11], and 62.8% in [13]. Because an exact comparison is difficult across different technologies, we implemented one of the prior designs (the CSHM-based DCT design from [13]) and compared it with a DCT design based on our techniques. Results show that TERRA techniques can achieve substantially lower energy for an image quality of about 30dB.

We also anticipate that our strategy is extendable to a larger class of algorithms. Our approach does not require changing the algorithm itself, e.g. to allow for early termination. Instead, we directly re-design the implementation to tolerate timing errors. Another difference with [11], [12], [13] is that their approach only allows a discrete set of quality-energy points. By contrast, our technique enables a range of trade-offs along a continuous quality-energy profile.

In earlier version of this work [1], we presented basic techniques to allow tradeoffs between quality and energy. Here, we extend this work by: (a) substantially expanding formal analysis of design choices that need to be made in implementing the timing-error accepting strategy, and (b) presenting novel post-processing techniques to improve the quality of images produced by our error-accepting circuits.

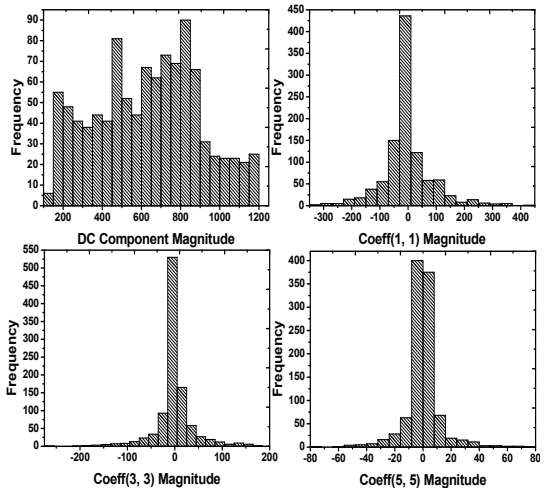


Fig. 1. Frequency distribution of IDCT coefficients for sample image.

A widely used post-processing technique is filtering, such as a 2-D median filter. In [15], median filtering is used to remove noise. In this work, we implement a simplified median filter that can quickly estimate the median of an array of pixels, such that the computational complexity is reduced and a low-energy design is achieved. Another existing approach for error reduction is to identify the erroneous results and then replace them with an approximated one [7]. We propose an image filter with error limiting that performs a partial substitution on the output pixel instead of replacing all of its bits. This significantly simplifies the error checking and correction logic. In contrast to previous work, our focus is on energy minimization under performance constraints instead of pure performance or throughput optimization [15].

III. Image Processing

The 2D-IDCT and 2D-DCT computations can be represented by $I = C^T \cdot A \cdot C$ and $I = C \cdot A \cdot C^T$, respectively, where C is the orthogonal type-II DCT matrix and A is the spectrum coefficient matrix. It is customary to implement the 2D-IDCT/DCT as a sequence of two 1D-IDCT/DCTs. For each 1D-IDCT/DCT, the core algorithm is a matrix-vector dot product. For IDCT, the transformation is:

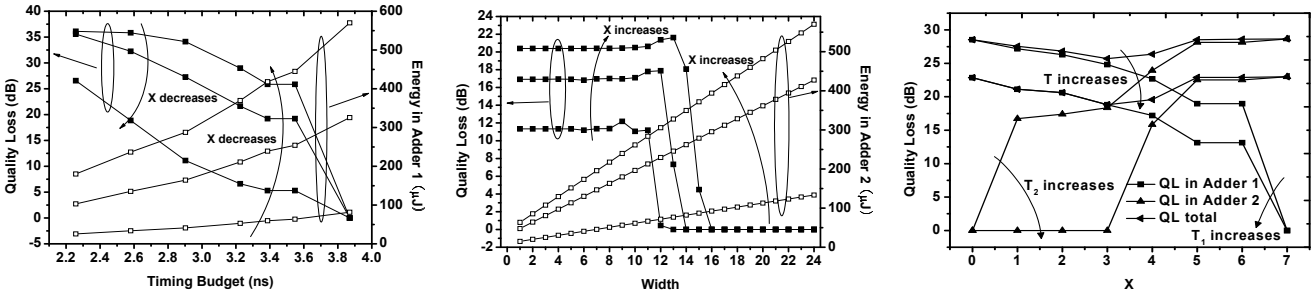
$$T(k) = \frac{c(k)}{2} \cdot \sum_{n=0}^{N-1} x(n) cos[\frac{(2k+1)n}{2N}\pi]$$

$$N = 8, c(0) = 1/\sqrt{2}, c(k) = 1, 0 \leq k \leq N-1$$

where $x(n)$ is the data being processed. The DCT is very similar, except that the coefficient matrix is transposed. The following discussions will focus on a 2D-IDCT. Application to a corresponding 2D-DCT will be discussed later.

A. Error control through knowledge of operand statistics

When V_{DD} is scaled down, large magnitude timing errors are very likely to happen in additions of small numbers with opposing sign. Such additions lead to long carry chains and are the timing-critical paths in the adder. The worst case for carry propagation occurs in the addition of -1 and 1. In 2's



(a) Energy and quality loss in Adder 1 (b) Energy and quality loss in Adder 2 (c) Quality loss vs. component classification

Fig. 2. Quality-energy tradeoffs in Adder 1 and Adder 2.

complement representation, this operation triggers the longest possible carry chain and, thus, experiences timing errors first. Crucially, when a timing error occurs, the apparent result will also have a very large possible numerical error due to carry propagation into the MSBs leading to a large magnitude mismatch compared to the error-free result. For example, in an 8-bit computation, the error magnitude can be up to 128. This analysis and this problem is, of course, specific to the 2's complement representation of signed numbers. However, our techniques can also be used in sign-magnitude representation. As will be detailed later, in sign-magnitude arithmetic, subtractions or opposing-sign additions are internally computed using 1's or 2's complement logic. This results in similar timing error behavior and our techniques remain effective.

In the 2D-IDCT algorithm, the additions that involve small-valued, opposite-sign operands occur in the processing of high-frequency components. This is because the first 20 low-frequency components contain about 85% or more of the image energy [13]. Hence, the magnitude of high-frequency components tends to be small, and coefficients follow a Laplace distribution with high probability densities concentrated in a narrow range [16], as shown in Fig. 1. Furthermore, the Laplace distributions are zero-centered, which implies that high frequency components also tend to have opposing signs. As such, a significant amount of quality loss at scaled V_{DD} can be attributed to additions involving such components. The first specific technique we employ is based on the realization that an adder with a bitwidth smaller than required by other considerations can be used to process such operands. Two objectives are achieved by using such adders: the magnitude of quality loss is reduced and its onset is delayed. Large-valued operands, of course, require a regular-width adder. Note that in an actual implementation it is possible to utilize a single adder with variable bitwidth.

In the IDCT algorithm, the classification of matrix elements can be done at design time. This raises the question of (a) how to best perform this classification; and (b) how to identify the optimal bitwidth of the reduced-width adder. In the following, we develop a model to enable such a design optimization. We define Adder 1 as the regular-width adder and Adder 2 as the reduced-width adder. In classifying the components, we seek to find the boundary, within the data matrix, between the upper-left low-frequency components and the lower-right high-frequency components (Fig. 3). For the convenience of analysis, instead of evaluating the impact of V_{DD} scaling on delays, we work with an equivalent reduction in available timing budget. We therefore define the following parameters of our model:

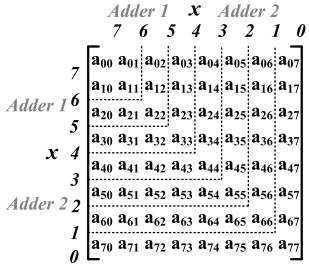


Fig. 3. Partitioning of input matrix.

x	Boundary between high-/low-frequency coefficients.
D_1	Worst-case delay of Adder 1.
D_2	Worst-case delay of Adder 2.
T_1	Timing budget of Adder 1.
T_2	Timing budget of Adder 2.

Here, timing budgets T_i are defined as the clock periods under which adders operate. Based on this notation, we can study the Q-E characteristics of the two adders under scaled V_{DD}. By exploring adder characteristics, we are able to identify the optimal partitioning strategy from the point of view of achieving a globally optimal Q-E result. We assume throughout this discussion that $T_2 = D_2$, i.e. that no timing errors are allowed to occur in Adder 2. Furthermore, we assume a common budget $T = T_1 = T_2$, which implies that both adders are affected by V_{DD} scaling in an identical manner.

We first study the Q-E relation for the regular width adder, shown in Fig. 2(a). The right axis shows the energy value at different timing budgets T_1. As expected, allotting a smaller timing budget, which entails an equivalent lowering of V_{DD}, results in a reduction of energy. Increasing the number of matrix components processed in the reduced-width adder, i.e. increasing x, results in fewer additions performed by Adder 1, and thus a lower energy at the same timing budget. The quality loss (shown on the left axis) is initially low when the allotted timing budget is high and few computations experience error. As T_1 is reduced, however, we begin to observe that the quality loss is smaller for larger x. This corresponds to the scenario in which fewer operations are performed by Adder 1, and thus there is less opportunity for timing errors to occur.

The Q-E behavior of the reduced-width adder is shown in Fig. 2(b). We are specifically interested in finding the Q-E behavior as a function of the bitwidth. Note that because no *timing* errors are allowed in Adder 2, an exploration with respect to timing budget, as shown for Adder 1 above, would have no purpose. We see that for large bitwidths of Adder 2, there is no quality loss. A significant reduction in quality occurs with the onset of overflow errors when the magnitude of data being processed is larger than the available adder width.

The analysis of the system Q-E behavior combines the

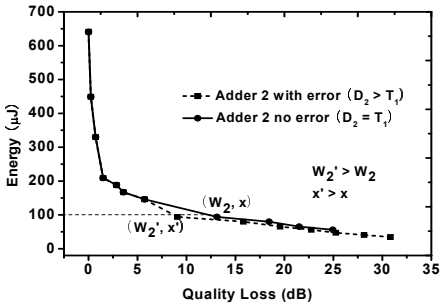
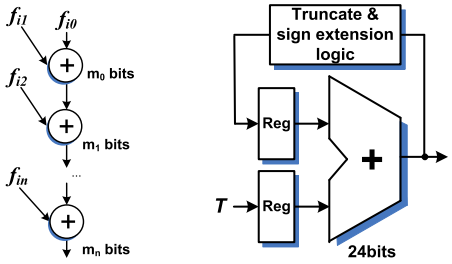


Fig. 4. Energy vs. quality loss Pareto front - comparison.



(a) Technique abstraction (b) Implementation

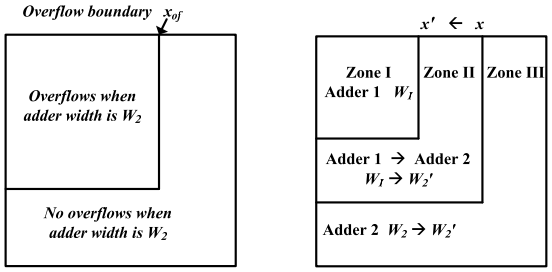
Fig. 6. Reduced width adder.

behavior of Adder 1 and Adder 2. This enables exploration of the x, D_2, W_2, and T_1 design space in order to find an optimal Q-E solution. The primary trade-off involves the choice of x. From Fig. 2(c), we can see that the total quality loss reaches a minimum when x is around 4. For larger values, the quality loss due to Adder 2 becomes excessive. For smaller values, the quality loss is dominated by errors from Adder 1. However, the optimal choice of x also depends on both the total timing budget available as well as the bit-width of Adder 2. The set of optimal design decisions is best represented as a Pareto curve in the energy-quality space as shown in Fig. 4. The figure shows the Pareto points, i.e. $min(Q|E)$, that are generated by different choices of x and W_2 at different T_1.

To understand the behavior in Fig. 4 and trace the dependence of the optimal x on T_1 and W_2, we first study the simple case when $D_2 \leq T_1$. Then, we relax the constraint to allow $D_2 > T_1$, and we adjust x and W_2 under a fixed T_1 to determine the new optimal set of x and W_2.

Under the constraint that $D_2 \leq T_1$, we can observe that: 1) the optimal x is set by the overflow boundary (x_{of}); and 2) the optimal Adder 2 width is the maximum Adder 2 width (W_{2max}), which is set by T_1. The x_{of} here is defined as the maximum possible x for a given W_2 without having overflows in Adder 2, as shown in Fig. 5(a). For a given timing budget T_1 and $D_2 \leq T_1$, there must not be any timing errors and we can define a maximum Adder 2 width as W_{2max}. Since the onset of overflow immediately leads to large errors (Fig. 2(b)), W_{2max} also sets a maximum x of x_{of} at the boundary at which overflows appear. At the same time, we always aim to send as much data as possible to the error-free reduced-width adder (Adder 2), so as to reduce timing errors in the full-width adder (Adder 1). Hence, we choose x to be at its maximum (x_{of}).

To further explore the design space beyond the point for which no timing errors are allowed in Adder 2, we can observe that in the absence of overflows, the output timing error in a



(a) $D_2 \leq T_1$ (b) $D_2 > T_1$

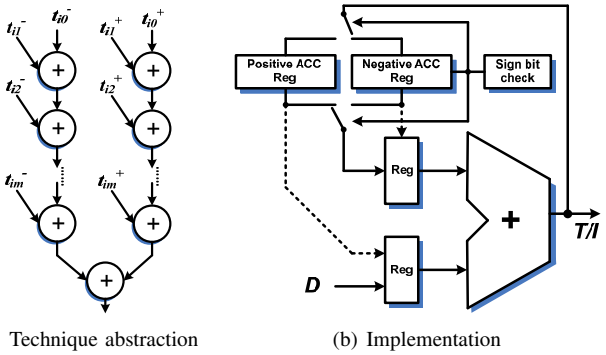
Fig. 5. DCT coefficient partitioning.

wide adder is greater than or equal to the error in a smaller-width adder when both adders process the same operands. Accordingly, sending more data to Adder 2, i.e. increasing x, will make it possible to further reduce the quality loss, even in the presence of timing errors in both adders. However, in order to avoid exceeding the overflow boundary with its large quality loss, we also have to increase W_2 and hence x_{of}, relaxing the timing constraint for Adder 2 to $D_2 > T_1$. As shown in Fig. 5(b), data in Zone I and Zone II is originally processed by Adder 1 using width W_1. Data in Zone III is processed by Adder 2 using width W_2. After increasing x, data in Zone II is processed by Adder 2 using width W_2', s.t. $W_2 < W_2' < W_1$. The quality loss in Zone II is reduced while the quality loss in Zone III increases. Since increasing x, i.e. sending more data to Adder 2, can reduce timing errors in Adder 1, but increasing W_2 leads to more timing errors in Adder 2, there exists an x (and W_2) with maximum quality loss reduction. These points correspond to the Pareto front of the dashed line in Fig. 4.

In the implementation, the reduced-width addition is actually realized using the truncated result of a regular-width adder sharing the same core logic. The combined adder architecture is shown in Fig. 6. The indices of the frequency coefficients are used by the control logic to determine whether to feed them into a full-width or reduced-width addition. The control logic compares the index of the matrix component currently being processed with the predetermined classification constant x. The output of this comparison is used to activate a truncation logic. The truncation logic takes a reduced number of LSBs from the full-width adder output according to the pre-designed Adder 2 width, sign extends them back to the full width, and feeds the result back into the destination accumulator.

B. Error control by dynamic reordering of accumulations

The technique introduced in Section III-A is able to delay the onset of large-magnitude errors in individual two-operand additions. The second technique presented in this section is based on a reduction of the cumulative quality loss resulting from multiple additions, such as accumulations, which are a key component and optimization target of many DSP algorithms [17], and, specifically, of the IDCT. The key observation in our context is that if positive and negative operands are accumulated separately and added only in the last step, the number of error-producing operations is reduced to one last addition that involves operands with opposite sign. At the same time, the operands involved in this last addition are guaranteed to be larger in absolute value than any individual opposite-sign



(a) Technique abstraction (b) Implementation

Fig. 7. Accumulation reordering architecture.

operands involved in the original sequence of additions. This guarantees that the reordered accumulation will result in a smaller quality loss under scaled timing.

Let us illustrate how the order of operations in accumulation affects the timing errors occurring at a given timing budget. As an example, consider four numbers (-1, 1, -1, 1) being accumulated. There are three possible sequences of accumulation:

Case 1: 11111111+00000001+11111111+00000001
Case 2: 11111111+11111111+00000001+00000001
Case 3: (11111111+11111111)+(00000001+00000001)

For case 1 and case 2, two of the additions have a large delay with a carry chain length of 8. For case 3, only the addition outside the brackets has large delay with a smaller carry length of 7. Hence, the total timing budget in case 3 is roughly half of that of case 1 and 2. Thus, we observe that the order of accumulation can significantly affect the frequency of worst-case delay as well as the length of the longest carry chain.

The proposed implementation uses the sign bits in the MSB to separate the positive and negative operands when loading data. The implementation is shown in Fig. 7. The control logic checks the sign bits and accumulates positive and negative numbers in separate accumulation registers. Then, in a final step, the results are added together. This final addition can in turn be protected against timing errors using either one of the techniques presented in Section III-A or III-C.

Compared to the original implementation, the reordered accumulation carries extra overhead for the reordering logic and duplicate accumulation registers. Nevertheless, simulation results show (Section V) that the technique can significantly improve the quality-energy profile under scaled timing.

C. Preventing error spread and amplification

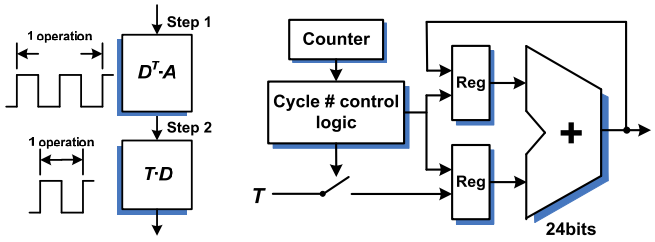
In previous sections, we presented techniques for targeting individual error sources at the operation and block level. With knowledge of the application, we now further focus on control of sources of errors that have the potential to be spread and amplified at the algorithm level. More specifically, we propose a technique using algorithm-level retiming to explicitly prevent errors in critical steps that may have a large impact on downstream results and hence overall quality. Similar retiming techniques have been applied dynamically in the context of error avoidance (based on error prediction) under timing speculation [18]. By contrast, we utilize static retiming to minimize errors under a fixed latency constraint in an overall error-accepting framework.

For the 2D-IDCT algorithm, analysis of control and data flow is relatively simple because it consists of two nearly-identical steps: $T = C^T \cdot A$ and $I = T \cdot C$. We address the problem of a timing error in Step 1. Such an error can generate multiple output errors in I because each element of T is used in multiple computations of Step 2. We can model this behavior by introducing an error matrix E, which is added to T such that the two algorithm steps become: $T' = T + E$ and $I = T \cdot C + E^{'}$. Here, $E^{'} = E \cdot C$ is the final error. Although E may have only one non-zero entry, the matrix product results in up to $size(A)$ errors vertically or horizontally in $E^{'}$. As a result, the noise in the decoded image of an unmodified IDCT has a stripe pattern (see Fig. 13 in Section V).

Thus, to avoid such wide-spread quality loss, we need to ensure that no errors occur in Step 1. We assume an architecture in which supply voltage can only be scaled uniformly. We now consider algorithm-level timing budgets allocated to a sequence of operations. Hence, timing budgets refer to the number of cycles at a given clock period. If timing budgets are allocated to steps based on worst-case analysis, any reduction in V_{DD} would lead to a reduced timing slack in Step 1 and hence un-allowable levels of errors being generated there. We therefore propose a strategy to allocate extra timing margins to critical steps, such as Step 1. Importantly, given overall latency constraints for the design, as is the case for many real-time image or video coding applications, end-to-end algorithm timing must remain constant and performance must not be degraded. Thus, an important element of protecting the early algorithm steps is a re-allocation strategy that shifts timing budgets between steps. Maintaining a constant total time, we show how to borrow computing time from non-critical algorithm steps in order to increase timing margins in critical ones, all while reducing overall quality loss.

To implement such a strategy, we make the timing budget in each step adjustable. The original minimum error-free timing budget for each step is: $T_{step1} = N_1 \times T_{clk}$ and $T_{step2} = N_2 \times T_{clk}$, where T_{clk} is the clock period, and N_1 and N_2 are the number of cycles in each step. In the original 2D-IDCT, steps are identical and $N_1 = N_2 = N$. To adjust the budget, we need to divide it into multiple parts. A division factor M (M is greater than 1, and it is an integer) is used to make $T_{step1} = NM \times T_{clk}/M$, and $T_{step2} = N \times T_{clk}/M$. V_{DD} is then scaled down, increasing the propagation delays. Consequently, T_{clk} is scaled to $T^{'}_{clk}$ such that $2N \times T_{clk}$ is equal to $NM \times T^{'}_{clk}/M + N \times T^{'}_{clk}/M$, i.e. $T^{'}_{clk} = 2T_{clk}/(1+1/M)$. Hence, the new clock frequency is: $f^{'}_{clk} = T^{'}_{clk}/M = 2/((M+1)T_{clk})$. Since the total budget is fixed, we disproportionally shift timing budgets under scaled V_{DD} from Step 2 to Step 1. Note, however, that the factor M cannot become too large. Otherwise, the clock frequency would be too high and timing errors would not remain restricted to the adder in Step 2.

The implementation includes logic to allocate different timing budgets to each step (Fig. 8). We empirically choose M to be 2 and increase clock frequency accordingly. The control logic includes a 1-bit counter to keep track of the cycle counts for each step. In Step 1, each operation is assigned 2 cycles, while each operation in Step 2 is assigned 1 cycle.



(a) Technique abstraction (b) Implementation

Fig. 8. Rescheduling of algorithm steps.

IV. Image Post-Processing to Mitigate Errors

Techniques discussed so far have dealt with preventing or minimizing errors in the output image. While the described techniques significantly reduce energy at an acceptable PSNR, they result in some undesired localized visual artifacts that significantly degrade *perceived* image quality. The reason is that good PSNR alone is not a guarantee of acceptable visual quality of the image.

In the following, we develop energy-efficient means of reducing such image artifacts for the 2D-IDCT design. Artifacts can be divided into two categories: salt-and-pepper noise and stripe artifacts. Salt-and-pepper noise is a pattern of randomly occurring white and black pixels. In our 2D-IDCT system, this type of artifact is caused by timing errors in step 2 (Fig. 8(a)). By contrast, stripe artifacts are error patterns appearing as black and white lines of pixels, and are produced when timing errors happen in step 1 of the IDCT: errors in step 1 are amplified through the matrix multiplication in step 2, resulting in a stripe shape. Depending on the multiplication order, the stripe can be vertical or horizontal. In our algorithm, we perform an operation $T \cdot D$, where errors in the intermediate matrix T spread across different rows in the final output, resulting in a vertical stripe. If the order of two steps is reversed, artifact stripes would be horizontal. This property will affect the implementation of post-processing techniques. Importantly, because the logic for post-processing is quite simple we are able to guarantee that its delay is less than the adder delay. In this way, we are certain that post-processing logic is free of timing-induced errors.

To reduce the artifacts, we propose two separate filtering techniques. These two techniques can be implemented individually, or combined together. In this work, we demonstrate how to implement them separately.

A. Median Filtering

The first technique is median filtering. The algorithm uses a sliding window to replace each entry with the median of its neighboring entries. While preserving edges, a median filter is effective at removing localized high-frequency image artifacts, such as the aforementioned salt-and-pepper and stripe distortions, when the noise level is low [19]. Compared to other filters, it is also less complex and only requires comparisons. Therefore, the hardware implementation of a median filter can be made simpler and more energy-efficient.

In our implementation, median filtering is performed on the converted output stream, and is applied to the entire image. In the design without median filtering, the output image is stored in memory and then sent out after each 8×8 block computation is done. With median filtering, each pixel is filtered when being sent out. As discussed before, stripe-shaped artifacts manifest themselves as single vertical lines. We therefore perform horizontal median filtering, which limits and localizes artifacts in the filter window. To minimize hardware overhead and maximize energy savings, we use the simplest possible 1-D median filter with window length 3. We further apply several optimizations to reduce hardware complexity and improve filter performance for our application.

First, a conventional filter checks the current pixel and all other pixels within the window to determine which one to output. Hence, the buffer needed for a length-3 median filter window is of size 2. However, in our experiments, the implementation of such a filter results in up to 10% area overhead. To further reduce complexity, we can develop an approximate median filter. The most visible artifacts are usually due to errors in MSBs of pixels. Hence, we can apply median filtering by comparing pixel MSBs only. Experiments show that the two most significant bits are sufficient to generate an output with only 0.3dB PSNR degradation compared to the case when all bits are used for filtering. This reduces area overhead to 1%.

Another drawback of a conventional median filter is that it will produce a modified output even in the absence of timing errors. To reduce this effect, we further modify the median filtering algorithm as follows: if the MSB of the median is the same as that of the unfiltered pixel, the filter will output the original pixel instead of the median. As such, if there is no or only a small timing error, the unfiltered output will be passed through. This modification is based on the observation that median filtering is effective at removing large single pixel outliers. Hence, if the MSB of the median is different from that of the unfiltered pixel, an outlier is likely detected and the filtered pixel is used to remove it. Otherwise, it means that there is no large outlier, and the unfiltered pixel is used. In this way, we can expect that with very high probability, both the LSB errors introduced by median filtering as well as the outliers can be reduced. The modified technique also significantly reduces the distortions caused by conventional median filtering. In the absence of timing errors, a conventional median filter leads to the loss of vertical lines. By contrast, our modified median filter preserves them.

Lastly, a third drawback in traditional median filters is the boundary issue. Pixels at the boundary of each 8×8 block can not be filtered because the data in the filter window is insufficient. To solve this problem, we use the following strategy to generate a window for the boundary pixels: assume that the pixel row being filtered is D_0, D_1, D_2, D_3...., where D_0 sits at the left block boundary. We then use the same window (D_0, D_1, D_2) for computing both D_0 and D_1. The problem with this windowing strategy is that for both D_0 and D_1, the filtered output is the same. This can cause visible vertical stripe patterns at 8×8 image block boundaries. However, such patterns are automatically mitigated by the previous technique, in which the filter, in most cases, will output the original pixel instead of the filtered one.

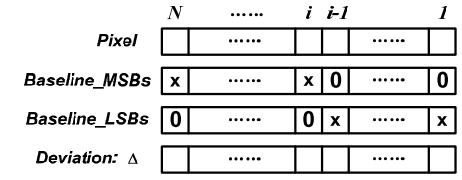


Fig. 9. Bitmap for all components.

B. Error Limiting

Our second proposed post-processing technique uses error limiting. It relies on the observation that typical images have low local spatial variations. Hence, pixels within each block are likely to have the same MSBs. In the frequency domain, this manifests itself as a large DC and small AC components. Based on this observation, each pixel in a 8×8 block can be represented as the sum of a baseline and a deviation. The baseline value is derived directly from the DC component. It is the same for all pixels in a block, and is the average over all 8×8 pixel values. The deviation is obtained from the AC components, and it differs from pixel to pixel.

In typical smooth image regions, deviations will be small and pixel values are likely to have similar values, i.e. the same MSBs as the baseline average. Under scaled V_{DD}, the baseline can be easily obtained without errors. In a 8×8 2D-IDCT, it is simply $1/8$ of the DC value, i.e. it can be computed by shifting the DC component by 3 bits. At the same time, pixels tend to have timing errors in their MSBs first. Therefore, if there are only small local deviations, we can substitute the pixel MSBs (which may be affected by timing errors) with the error-free baseline MSBs, limiting the impact of any timing errors. Because error checking incurs area and energy overhead, we want to perform such substitution blindly. To do so and not introduce additional errors, we have to ensure that baseline MSBs are a correct predictor of error-free pixel MSBs, i.e. that $Pixel[N:i] = Baseline[N:i]$. The question therefore becomes 1) when this equality holds, and 2) if it holds, for what values of i.

We address these questions by looking at a pixel representation as follows:

$$\begin{aligned} Pixel &= Baseline + \Delta \\ &= Baseline_{MSBs} + Baseline_{LSBs} + \Delta, \end{aligned}$$

where $Baseline_{MSBs}$ and $Baseline_{LSBs}$ represent zero-padded splits of $Baseline$ at the ith bit position, and Δ represents the deviation, see Fig. 9. We call $Baseline_{LSBs}+\Delta$ the residue term.

To guarantee $Pixel[N:i] = Baseline[N:i]$ for a given i, we need to ensure that the MSBs (bits N:i) of the residue term are zero, i.e. that $0 \leq Baseline_{LSBs} + \Delta \leq 2^i$. We can rewrite this inequality as:

$$-\Delta \leq Baseline_{LSBs} \leq 2^i - \Delta \tag{1}$$

To find the i that satisfies inequality (1), we need to know both $Baseline_{LSBs}$ and Δ. We now show how to estimate both terms for the 2D-IDCT algorithm. To determine Δ, we can rewrite the 2D-IDCT algorithm, plug in the coefficients and separate the baseline and deviation terms:

$$\begin{aligned} x_{i,j} &= \frac{c(i,j)}{2} \cdot \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} C_{u,v}X_{u,v} \\ &= \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} C_{i,j}\cos[\frac{\pi u}{16}(2i+1)]\cos[\frac{\pi v}{16}(2j+1)]X_{u,v} \\ &= C_{0,0}X_{0,0}\cdot\cos[\frac{\pi 0}{16}(2i+1)]\cos[\frac{\pi 0}{16}(2j+1)] + \Delta \end{aligned}$$

The first term is the baseline:

$$\begin{aligned} Baseline &= C_{0,0}X_{0,0}\cdot\cos[\frac{\pi\cdot 0}{16}(2i+1)]\cos[\frac{\pi\cdot 0}{16}(2j+1)] \\ &= \frac{1}{8}X_{0,0} \end{aligned}$$

As mentioned before, from this, we can see that the baseline value can be simply computed by shifting the DC component ($X_{0,0}$). The deviation Δ becomes:

$$\Delta = \frac{1}{4}\sum_{\substack{v\neq 0 \\ if\ u=0}}^{7}\sum_{\substack{u\neq 0 \\ if\ v=0}}^{7} C_{i,j}\cos[\frac{\pi u}{16}(2i+1)]\cos[\frac{\pi v}{16}(2j+1)]X_{u,v}$$

Let t represent the upper bound of $|X_{u,v}|$, i.e.:

$$|X_{u,v}| \leq t \tag{2}$$

and

$$|\Delta| \leq C \cdot t, \tag{3}$$

where C is the following constant:

$$C = max_{i,j}(|\frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} C_{i,j}\cos[\frac{\pi u}{16}(2i+1)]\cos[\frac{\pi v}{16}(2j+1)]|)$$

We now rewrite (1) as:

$$-\Delta \leq C\cdot t \leq Baseline_{LSBs}$$
$$Baseline_{LSBs} \leq 2^i - C\cdot t \leq 2^i - \Delta$$

to derive a tighter bound for $Baseline_{LSBs}$:

$$C\cdot t \leq Baseline_{LSBs} \leq 2^i - C\cdot t \tag{4}$$

Again, this is the inequality that needs to be satisfied to guarantee that $Pixel[N:i] = Baseline[N:i]$.

We can perform error limiting based on inequalities (2) and (4). We first partition t into five regions and pre-calculate the corresponding values of $C\cdot t$ and $2^i - C\cdot t$ for different i. At runtime, we check the AC components $X_{u,v}$ of the 8×8 input block to determine a smallest upper bound t. Using the pre-computed bounds for the given t and the $Baseline$ computed from the DC component, we then find the smallest i for which (4) holds. If there is such an i, we replace pixel bits [N:i] with their baseline equivalents. Otherwise, no substitution is performed.

To further improve the error limiting technique, we reduce the upper bound t to allow more bits to be substituted. In practice, the deviation only reaches the upper bound when all $|X_{u,v}|$s equal t and their signs are the same as the corresponding 2D-IDCT coefficients. This rarely happens. Therefore, we can choose a smaller t. Such tweaking may lead to mis-substitution, but it removes other severe timing errors. We empirically determine a practical value of t to use.

Finally, we improve error limiting performance by introducing an allowed and forbidden state: when (3) is violated, we

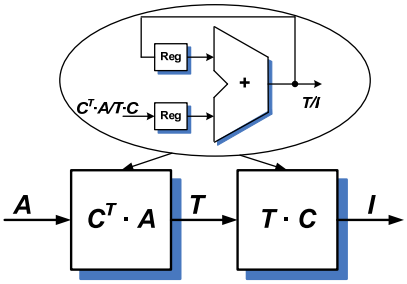


Fig. 10. 2D-IDCT design architecture.

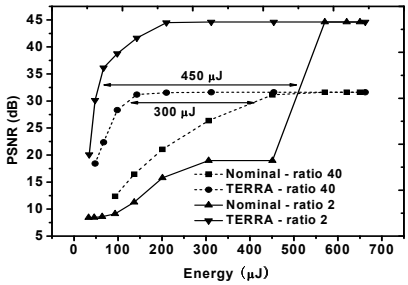


Fig. 11. Q-E tradeoff under different compression ratios.

still have knowledge what the MSBs of the pixels ought to be. For example, if the first three MSBs of the baseline value are binary 100, and if (3) is violated, but the value of $C \cdot t$ is still below a certain level, then the correct pixel MSBs can be 011 or 101, but definitely not 000 or 111. In such cases, we can remove the MSB error by defining 000 and 111 as forbidden states and using control logic to change these two states to either 100 or 011. Assuming the minimum difference between allowed and forbidden states is D, we introduce another bound $t' = D/C$, where C is the constant mentioned before. If all AC components $X_{u,v}$ are less than t', the MSBs within a given 8×8 block cannot take the value of a forbidden state. We use this concept to prevent the case when a white pixel becomes black and vice versa. We pick two sets of baseline MSB values (100 and 001), which are found to be most likely to have significant errors and determine the corresponding forbidden states for them. At run-time, we check for forbidden states and substitute baseline MSBs accordingly.

V. Experimental Results

We applied our techniques to 2D IDCT and DCT realizations. In a folded architecture [20], each 1D-IDCT/DCT shares the same physical, pipelined multiplier-accumulator (MAC) unit containing an adder and a multiplier, which minimizes the area of the whole design (Fig. 10). The test images are from the USC-SIPI image database [21]. Only the Y signal of each Y:Cb:Cr format image is used.

The IDCT data and coefficient matrices A and C have 16-bit and 8-bit resolution, respectively. By contrast, in the DCT case, both data and coefficient matrices have 8-bit resolution while the output resolution is 16-bit. The multiplier in the arithmetic unit is pipelined and has a width of 8×16 bits. The adder has a width of 24 bits and operates as an accumulator in the IDCT/DCT process. The error control techniques we introduce can be applied to various types of adders, and we realized them on a ripple-carry adder (RCA), a carry-select adder and a carry-lookahead adder. We conduct most of our experiments using a ripple-carry adder because it has better Q-E tradeoff at low energy compared to a tree adder, as we will demonstrate below. Starting from a balanced design that pairs a pipelined multiplier with a tree adder, we can replace the fast tree adder with a slow RCA. Under a constant timing budget, this leads to acceptable timing errors using our techniques at nominal voltage. Combined with a slightly faster and more complex multiplier, such a design restricts the timing errors entirely to the adder even when scaling voltage below nominal. As our results will show, our over-scaled implementation requires less energy than a balanced design that uses a fast adder to achieve the same quality and performance.

A 2D-IDCT block is usually used in an image decompression system in which input data is quantized. We therefore added a quantization step to generate realistic 8×8 IDCT input data. The quantization table is taken directly from the JPEG standard (Table $K.1$). We experimented with different compression ratios to test the effectiveness of proposed techniques (Fig. 11). In these experiments, the ratio is defined as the total number of bits required for the original images divided by the total number of bits required for quantized DCT data. A high compression ratio means that the numbers in the quantization table are large and more high frequency components are reduced to zero. At a low compression ratio of 2, the impact of quantization on the effectiveness of proposed techniques is not noticeable. At a compression ratio of 40, the initial quality is lower, as is expected. However, the rate of quality degradation is also lower. This is because aggressive quantization leads to many high-frequency DCT coefficients becoming zero, which reduces the likelihood of timing errors due to addition of small opposing-sign operands. Despite this intrinsic benefit of quantization, the proposed techniques are effective even at high compression ratios. This is because there are still many non-zero entries left in the DCT matrix. Experiments show that the proposed techniques with quantized data achieve about 60% to 80% energy savings for various levels of quantized data. In the following discussions, we use quantized data with a compression ratio of 40 to measure the achieved energy savings.

The 2D-IDCT/DCT designs are implemented in Verilog-HDL and synthesized using Design Complier with the OSU 45nm PDK. In the IDCT case, the design includes dequantization and clipping steps. In early versions of this work [1], we used HSPICE to estimate the delay and energy of single gates and fitted energy and delay models for voltage scaling. To further improve evaluation accuracy, we now run SPICE-level simulations on the whole design. We use Synopsys Hercules to translate the RTL code into a SPICE netlist and then build a NanoSim + VCS testbench to obtain final output images and energy-delay results through RTL and SPICE simulations, respectively. IDCT and DCT follow the same computational process. Other than removing dequantization and clipping steps, using different coefficients and reversing the partitioning of computations along output instead of input matrices, the same TERRA circuit is applied in both cases. In the following, unless otherwise specified, we first show results for the 2D-IDCT case and then extend those to a 2D-DCT.

TABLE I
ENERGY SAVING AND AREA OF IDCT.

	V_{DD}	Energy Saving	Area μm^2	Delay@1.1V
Original	1.1	0%	119337	3.88 ns
Reduced-width	0.95	47.9%	122930	3.91 ns
Reorder	0.95	36.7%	125237	3.91 ns
Re-budget	0.95	43.2%	122140	3.88 ns
All three	0.90	59.0%	125023	3.91 ns

TABLE II
ENERGY UNDER DIFFERENT COMBINATIONS.

Component	0	1	2	3	4	5
Eng (μJ)	463	357	267	279	287	296

A. Baseline IDCT and DCT Designs

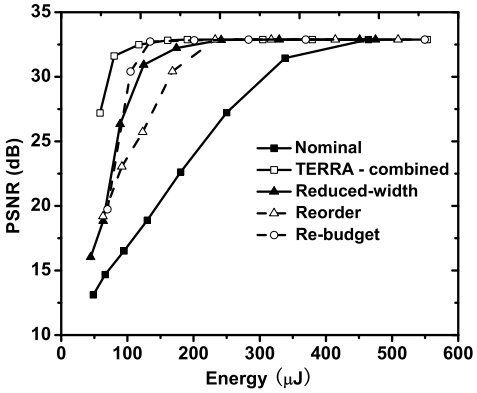
Table I shows the energy savings for each technique and their combination when applied to an IDCT. Energy savings are computed at PSNR = 30dB with the processing rate being a constant 11ms per 256×256 frame.

Individual techniques can be combined to achieve maximum energy savings. However, since the described techniques all have varying impact on the different frequency components, their optimal combination is not obvious. Using the technique of Section III-C, a larger timing budget is given to the earlier algorithm step. This change impacts all frequency components. On the other hand, the technique of Section III-A impacts mainly the high-frequency components (since they are the components that involve small-valued operands). Finally, the technique of Section III-B impacts operands with opposing sign, no matter if they are low- or high-frequency components.

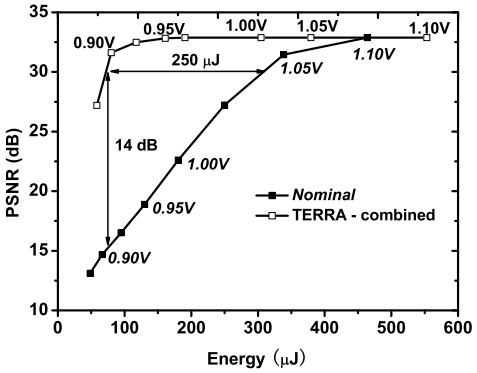
Based on these observations, we devised the following strategy for selectively applying techniques to different algorithm steps and frequency components: (1) In Step 1, we allocate more cycles only to the low-frequency components while using dynamic reordering and a reduced-width adder to process the high-frequency components; (2) In Step 2, timing errors are not propagated into later steps, so only the reduced-width adder and dynamic reordering are applied. In this combination, the total number of clock cycles needed in Step 1 is smaller than what the technique introduced in Section III-C would require to achieve the same quality level. Hence, under a fixed total time, the adjusted clock period T'_{clk} is larger and there exists more timing slack for energy savings.

The key problem is to determine which low-frequency components in Step 1 require more cycles after applying techniques from Sections III-A and III-B. Since the size of the frequency coefficient matrix in a 2D-IDCT is small, we can do a brute-force exploration to determine the best assignment. Table II shows the results of such simulations. Results indicate that the smallest energy is obtained when allocating more time (two cycles in our implementation) to the computation of the first two low-frequency components.

The PSNR vs. energy profiles for individual and combined techniques are shown in Fig. 12. A significantly improved Q-E trade-off is generated by a non-trivial combination of individual techniques. Finally, a set of sample images under scaled V_{DD} is shown in Fig. 13. Note that achieving a similar energy reduction by conventional V_{DD} scaling would result in unacceptable degradation of image quality (Fig. 13(b)). To further demonstrate the effectiveness of our TERRA techniques,



(a) Individual PSNR vs. energy profiles



(b) Combined PSNR vs. energy and voltage profile

Fig. 12. 2D-IDCT quality-energy profiles.

(a) Nominal: Energy=463μJ V_{DD}=1.1V PSNR=32.9dB

(b) Nominal: Energy=160μJ V_{DD}=0.95V PSNR=20.7dB

(c) TERRA: Energy=161μJ V_{DD}=0.95V PSNR=32.8dB

(d) TERRA: Energy=117μJ V_{DD}=0.9V PSNR=32.4dB

Fig. 13. Image quality under different energy budgets.

we test the 2D-IDCT design on a larger set of images [21]. As shown in Fig. 14, for all the test images, our design can significantly reduce the probability of large timing errors, and thus improve PSNR at smaller energy values.

Fig. 15(a) shows the effectiveness of our techniques at the level of individual bits when running at a supply voltage of about 0.95V. When no design optimizations are applied, the 2D-IDCT output has severe MSB errors (Fig. 15(a)). After

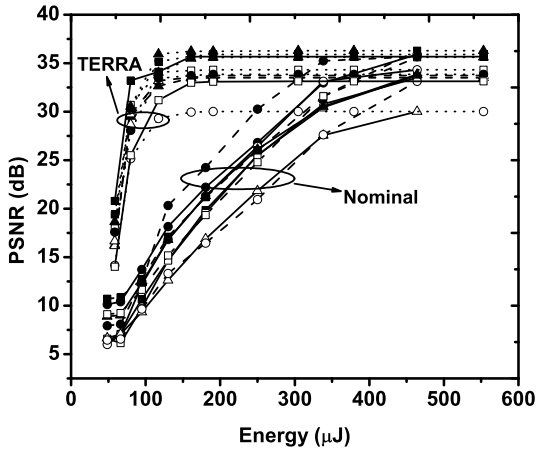
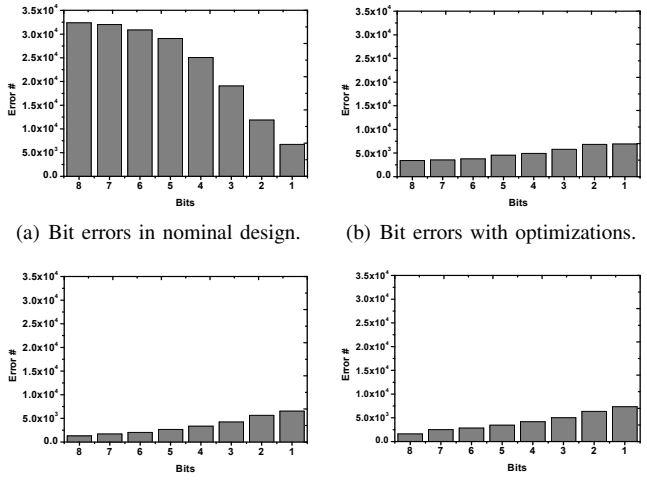


Fig. 14. Q-E curves for various images with/without our techniques.



(a) Bit errors in nominal design.

(b) Bit errors with optimizations.

(c) Bit errors with optimizations and error limiting.

(d) Bit errors with optimizations and median filtering.

Fig. 15. Frequency of errors in individual bit positions.

applying the three optimizations (Fig. 15(b)), both MSB and LSB errors are reduced. However, MSB errors are reduced much more significantly than LSB errors. Overall, pixel errors at the IDCT output are reduced from an unoptimized 40% down to 0.3% using our techniques.

To better quantify the effectiveness of this work, we compare the achieved energy savings with those produced by alternative approaches to approximate implementations of image processing circuits. For that, we also applied TERRA techniques to a 2D-DCT design. We implemented an approximate 2D-DCT design using the computation sharing multiplication technique (CSHM) described in [13]. Both designs use the same folded architecture described before, while the CSHM-based 2D-DCT replaces the pipelined MAC unit with a CSHM arithmetic unit. The two designs are synthesized using the same 45nm OSU library, and they are compared at identical performance. Results in Fig. 16 show that the initial quality allowed by the CSHM design is slightly lower than in our implementation. This is due to the coefficient restructuring performed in CSHM. Under scaled voltage, the CSHM design discards long but less important paths, i.e., the ones for high frequency components. Since the DCT data in our experiments is quantized, many high frequency components are already

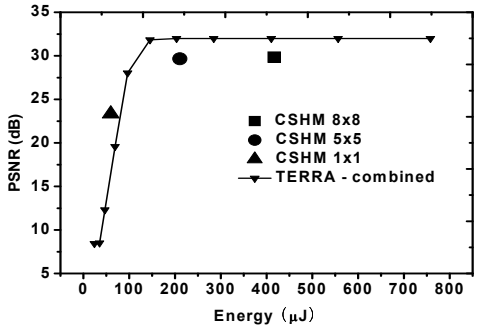


Fig. 16. Approximate 2D-DCT design: comparison of this work and CSHM.

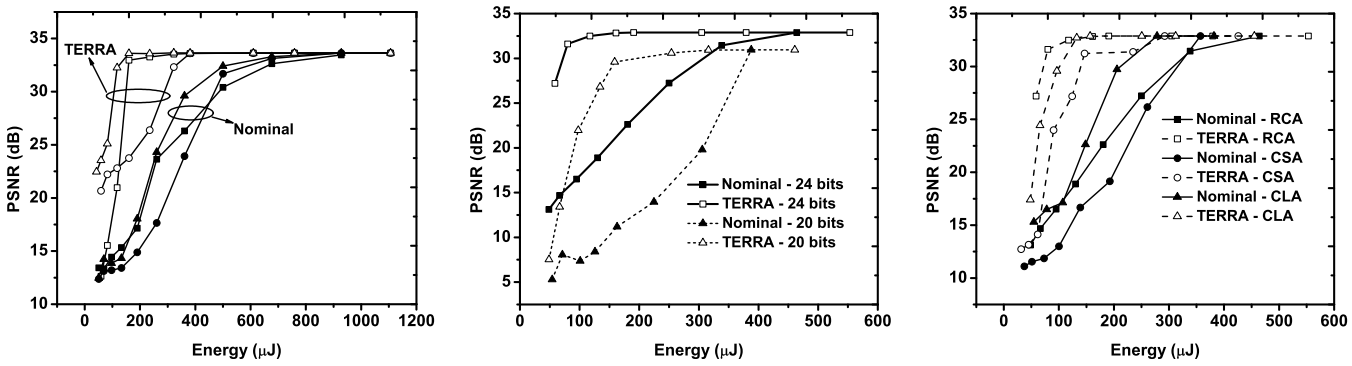
zero (which equals to discarding them). As a result, we can see that the quality difference between the CSHM-based design, which computes all 8×8 entries and the design which computes only the top-left (5×5) entries is negligible. Results suggest that the energy savings with the CSHM technique are about 50% at a quality of around 29dB, while our TERRA techniques can save about 71% energy on a 2D-DCT. At lower quality, however, the CHSM design performs slightly better.

B. Implementation Variants

In the following, we investigate the effect of several datapath variants on timing errors in an IDCT design with all optimizations applied. There are other existing design approaches that may already be able to similarly reduce timing errors and/or energy. We aim to study the effectiveness of our techniques in the presence of and compared to such alternatives.

First, in the discussion so far, TERRA techniques have been implemented using 2's complement data representation. Experiments demonstrate that they are also effective in systems based on sign-magnitude representation, which inherently separates operands of opposing sign causing earliest and largest timing errors. We implemented three sign-magnitude 2D-IDCT systems. The pipelined MAC units in these three implementations are designed based on different methods for performing sign-magnitude addition [22]: (1) translating sign-magnitude data to a 2's complement representation and using a regular adder to perform operations; (2) using a separate subtractor to handle opposing-sign additions, which internally are realized in 2's complement logic; and (3) employing a 1's complement subtractor to add opposing-sign numbers. Simulation results in Fig. 17(a) show that by applying TERRA techniques, about 37% to 60% energy saving can be achieved at a quality of 30dB. This is because even in sign-magnitude representation, additions of small opposite-sign numbers trigger the longest carry or borrow propagation in the adder or subtractor, respectively. Furthermore, due to the overhead for conversion, sign-bit logic and additional subtractors, the base energy of sign-magnitude systems is higher.

A widely-used, competing energy-saving technique is to reduce internal data precision through quantization. To compare the effectiveness of different strategies, we implemented a 20-bit 2D-IDCT design and compared its energy efficiency with our original 24-bit 2D-IDCT. In the 20-bit design, we employed a single MAC unit with 20-bit bitwidth. Input data and coefficients are 16-bit and 8-bit, respectively, while



(a) Sign-magnitude representation (b) 20 vs. 24 bits precision (c) Adder architectures

Fig. 17. IDCT designs with different implementation variants.

internal operations are truncated to 20 bits. Simulation results are shown in Fig. 17(b). Our optimizations are independent of the selection of a particular precision and we applied them to both designs. Results show that by using our techniques, about 59% energy savings can be achieved for a 20-bit design. Importantly, we find that the 24-bit TERRA design has uniformly better quality and energy than the reduced-precision implementation. We also observe that a nominal 24-bit design consumes less energy than a 20-bit design at the same quality level. This is due to the specific data patterns in the IDCT: the input data is quantized and many of the high frequencies are eliminated, which reduces the occurrence of early timing error offenders when voltage is scaled down.

The experiments so far relied on a ripple-carry adder. We now compare the error-energy behavior of different adder architectures, specifically, of a ripple-carry adder (RCA), carry-select adder (CSA) and carry-lookahead adder (CLA), see Fig. 17(c). As expected, we find that the 2D-IDCT designs using CLA, and to a lesser extent CSA, have smaller base energy if timing errors are not permitted. Since CLA and CSA have shorter critical paths, their initial energy advantages are due to our ability to reduce voltage more significantly without causing timing errors. Under scaled voltages, however, our techniques also enable significant energy savings of 54% and 50%, respectively, for CLA- and CSA-based designs. Our techniques are applicable because timing errors are still primarily caused by small operands. From Fig. 17(c) we further observe that the magnitude of energy reduction is largest for a RCA. As a result, while a RCA has a higher base energy, once timing errors are allowed, the RCA has lower energy than a CLA or CSA under equivalent performance and quality. The reason is due to the narrower, more balanced distribution of timing paths in the CLA or CSA [23], [24].

Our design so far relied on a fast pipelined multiplier that is paired with a slow adder, where the latter is overscaled using TERRA techniques to control timing errors under a common timing budget. By contrast, in traditional designs, either a fast, pipelined multiplier would be paired with a fast adder such as a CLA (balanced design 1), or a slow adder such as a RCA would be combined with a slower, non-pipelined multiplier (balanced design 2) to meet a certain performance goal. To compare design philosophies, we implemented both balanced approaches. The resulting Q-E tradeoffs are shown in Fig. 18. Balanced designs have a lower timing-error free base energy. This is because we can exploit timing slack for additional energy savings. However, under a timing error acceptance strategy, the unbalanced design can be scaled even further while maintaining almost perfect quality. As indicated in Fig. 17(c), at least in some cases, a slower RCA scaled to the same performance achieves a lower energy than a design that balances slack by using a fast adder. Overall, a TERRA approach at 32dB results in 50% and 53% energy savings over balanced designs 1 and 2, respectively. Understanding the extent of applicability of this observation requires further work, however. Additionally, while this paper focuses on the adder only, in future work we plan to investigate timing error acceptance mechanisms in multipliers.

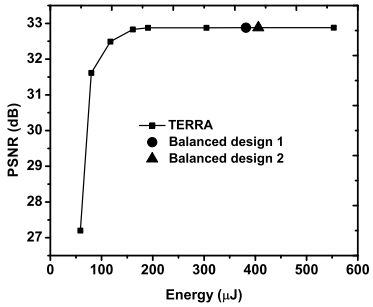
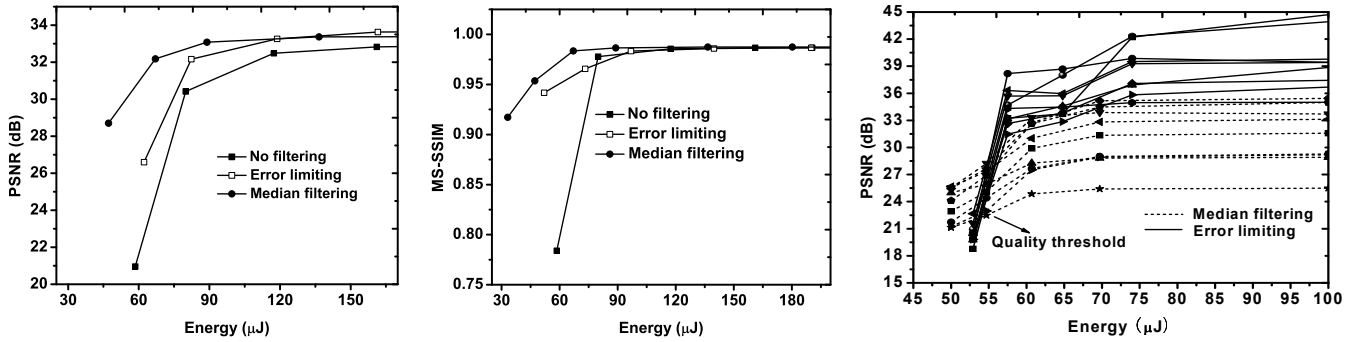


Fig. 18. Balanced/unbalanced designs.

C. Image Post-Processing to Mitigate Errors

In Section IV, we proposed two types of post-processing techniques to filter out 2D-IDCT artifacts. The combination of post-processing with error shaping optimizations is straightforward. For the median filtering technique, output data is buffered in memories with a size equal to the filtering window length and the approximate median of one data window is selected as the filtered result. This process operates directly on the output data and is independent from the error shaping techniques. Fig. 15(d) shows the bit-level error count of a whole image after using median filtering. Compared to Fig. 15(a), we can see that median filtering is effective in further reducing the MSB errors in the output image. However, it also leads to a slight increases in LSB errors.

Post-processing via error limiting involves both input and output data. Input data is analyzed to determine whether value limiting should be applied and how many MSBs should be affected. Output MSBs are subsequently overwritten if the control logic on the input side triggers the substitution. This technique is also independent of core error shaping techniques. As shown in Fig. 15(c), error limiting reduces MSB errors without increasing errors in the LSBs.



(a) PSNR curve (b) MS-SSIM curve (c) PSNR for various images

Fig. 19. Comparison between median filtering and error limiting.

TABLE III
AREA/ENERGY OF IDCT WITH POST PROCESSING.

Type	No postprocessing	Error Limiting	Median Filtering
Engergy (μJ)	117	140	137
Area (μm^2)	125023	125031	126443
PSNR (dB)	32.4	33.3	33.4
MS-SSIM	0.9857	0.9860	0.9866

Area, energy and image quality of the post filtering techniques are shown in Table III, where energy and quality is measured at the same voltage level as in Fig. 13(d) (PSNR of around 32dB). Corresponding images after post-processing are shown in Fig. 20(a) and Fig. 20(b). Both filtering techniques introduce a slight area and energy overhead. However, their 1 to 2dB PSNR improvement is larger than what simple scaling of voltages to the same energy level would deliver in an unfiltered design. Fig. 19(a) shows the PSNR vs. energy profile of post-filtering techniques. Although image quality is generally improved, we can note that at high energy levels, the PSNR curve for median filtering is worse than the unfiltered case. Yet, even at such high energy levels, the median filtered image looks as good as, if not better, than the unfiltered one. Furthermore, in intermediate regions, the PSNR metric is highly non-monotonic, which makes it difficult to fairly evaluate energy-quality tradeoffs. Therefore, we utilize an alternative multi-scale structural similarity (MS-SSIM) [25] metric, which is designed to accurately assess humanly perceived image quality. The MS-SSIM curve of different techniques under varying energy levels is shown in Fig. 19(b). MS-SSIM results confirm that, compared to the original case, both post processing techniques improve the perceived image quality over the whole energy range. This coincides with the visual appearance of the images (Fig. 20), which have less salt-and-pepper noise and look better. In addition, in MS-SSIM profiles, quality drops off at lower energy levels. As such, further energy savings can be achieved while maintaining an overall excellent image quality with an MS-SSIM > 0.90. Fig. 20(c) and Fig. 20(d) show resulting test images and energy levels.

To further determine which filtering method to use for a given application, we apply each of them to multiple images. The simulation results are shown in Fig. 19(c). There is a quality-energy tradeoff and break-even point for choosing between different filtering techniques. At high energy levels, median filtering introduces intrinsic errors and error limiting is better in terms of PSNR. But median filtering has a slightly

(a) Test image with error limiting: Energy=140μJ V_{DD}=0.9V PSNR=33.08dB MS-SSIM=0.9860

(b) Test image with median filtering: Energy=137μJ V_{DD}=0.9V PSNR=33.3dB MS-SSIM=0.9866

(c) Test image with error limiting: Energy=82μJ V_{DD}=0.875V PSNR=32.2dB MS-SSIM=0.9834

(d) Test image with median filtering: Energy=67μJ V_{DD}=0.875V PSNR=32.2dB MS-SSIM=0.9835

Fig. 20. Image quality after post-processing.

lower area overhead and outperforms error limiting when energy levels begin to drop. This effect is more pronounced when looking at MS-SSIM profiles, where perceptual base quality is less affected by the intrinsic smoothing introduced through median filtering. Overall, designers can select which post-processing technique to use depending on desired quality levels and energy budgets. The PSNR cross-over point between the two techniques depends on the image, and the threshold needs to be determined through simulations of representative images under scaled voltage. The quality required by a specific application scenario determines the technique to be employed.

VI. CONCLUSIONS

This paper presented techniques that enable architecture-level shaping of quality-energy tradeoffs under aggressively scaled V_{DD} through controlled timing error acceptance. We demonstrated these techniques on the design of a 2D-IDCT/DCT architecture. Results show that significant energy

savings can be achieved while maintaining a constant performance and good image PSNR. To further improve the visual quality, filtering techniques can be implemented to reduce visual image artifacts. In future work, we aim to generalize the proposed approach to other DSP applications, including use of optimized TERRA blocks in broader system contexts, such as full video coding, as well as integration into standard synthesis flows.

ACKNOWLEDGMENTS

This research was made possible in part by support from the National Science Foundation grant CCF-1018075. We would also like to thank Prof. Swartzlander at UT Austin for helpful discussions on the sign-magnitude MAC unit implementation.

REFERENCES

[1] K. He, A. Gerstlauer, and M. Orshanksy, "Controlled Timing-Error Acceptance for Low Energy IDCT Design," *Design, Automation and Test in Europe Conference and Exhibition*, pp. 492–499, 2011.

[2] S. H. Nawab, A. V. Oppenheim, A. P. Chandrakasan, J. M. Winograd, and J. T. Ludwig, "Approximate Signal Processing," *IEEE Journal of VLSI Signal Processing Systems*, vol. 15, pp. 177–200, 1997.

[3] J. T. Ludwig, S. H. Nawab, and A. P. Chandrakasan, "Low-Power Digital Filtering Using Approximate Processing," *IEEE Journal of Solid-State Circuits*, pp. 395–400, 1996.

[4] D. Anastasia and Y. Andreopoulos, "Software Designs of Image Processing Tasks with Incremental Refinement of Computation," *IEEE Workshop on Signal processing Systems*, pp. 249–254, 2009.

[5] A. Sinha and A. P. Chandrakasan, "Energy Efficient Filtering Using Adaptive Precision and Variable Voltage," *ASIC SOC Conference*, pp. 327–331, 1999.

[6] P. Albicocco, G. C. Cardarilli, A. Nannarelli, M. Petricca, and M. Re, "Imprecise Arithmetic for Low Power Image Processing," *46th Asilomar Conference on Signals, Systems and Computers*, 2012.

[7] R. Hedge and N. R. Shanbhag, "Soft Digital Signal Processing," *IEEE Transactions on VLSI Systems*, pp. 379–391, 2000.

[8] L. Wang and N. R. Shanbhag, "Low-power Filtering via Adaptive Error-Cancellation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 51, no. 2, pp. 575–583, 2003.

[9] T. Xanthopoulos and A. Chandrakasan, "A Low-Power DCT Core Using Adaptive Bitwidth and Arithmetic Activity Exploiting Signal Correlations and Quantization," *IEEE Journal of Solid State Circuits*, vol. 35, no. 5, pp. 740–750, 2000.

[10] F. Kurdahi, A. Eltawil, K. Yi, S. Cheng, and A. Khajeh, "Low-Power Multimedia System Design by Aggressive Voltage Scaling," *IEEE Transactions on VLSI Systems*, vol. 18, no. 5, pp. 852–856, 2010.

[11] J. Park, S. Kwon, and K. Roy, "Low Power Reconfigurable DCT Design Based on Sharing Multiplication," *IEEE Int'l Conference on Acoustics, Speech, and Signal Processing*, pp. III–3116–III–3119, 2002.

[12] G. Karakonstantis, D. Mohapatra, and K. Roy, "System Level DSP Synthesis Using Voltage Overscaling, Unequal Error Protection and Adaptive Quality Tuning," *IEEE Workshop on Signal Processing Systems*, 2009.

[13] N. Banerjee, G. Karakonstantis, and K. Roy, "Process Variation Tolerant Low Power DCT Architecture," *Design, Automation and Test in Europe Conference and Exhibition*, pp. 1–6, 2007.

[14] P. N. Whatmough, S. Das, D. M. Bull, and I. Darwazeh, "Circuit-level Timing Error Tolerance for Low-Power DSP Filters and Transforms," *IEEE Transactions on VLSI Systems*, 2012.

[15] R. L. Swenson and K. R. Dimond, "A Hardware FPGA Implementation of 2-D Median Filter Using a Novel Rank Adjustment Technique," *Int'l. Conference on Image Processing and Its Application*, vol. 1, pp. 103–106, 1999.

[16] E. Y. Lam and J. W. Goodman, "A Mathematical Analysis of the DCT Coefficient Distribution for Images," *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1661–1666, 2000.

[17] A. Chandrakasan and R. W. Brodersen, "Minimizing Power Consumption in Digital CMOS Circuits," *Proceedings of the IEEE*, vol. 83, no. 4, pp. 498–523, 1995.

[18] S. Ghosh, S. Bhunia, and K. Roy, "CRISTA: A New Paradigm for Low-Power, Variantion-Tolerant, and Adaptive Circuit Synthesis Using Critical Path Isolation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 11, pp. 1947–1956, 2007.

[19] E. Arias-Castro and D. L. Donoho, "Does Median Filtering Truly Preserve Edges Better than Linear Filtering?" *The Annals of Statistics*, vol. 37, no. 3, pp. 1172–1209, 2009.

[20] S. Uramoto, Y. Inoue, A. Takabatake, J. Takeda, and Y. Yamashita, "A 100-MHz 2-D Discrete Cosine Transform Core Processor," *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 492–499, 1992.

[21] USC SIPI Image Database. [Online]. Available: http://sipi.usc.edu/database/

[22] R. K. Richards, *Arithmetic Operations in Digital Computers*. New York: Van Nostrand, 1955.

[23] Y. Liu, T. Zhang, and K. K. Parhi, "Computation Error Analysis in Digital Signal Processing Systems with Overscaled Supply Voltage," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 18, no. 4, pp. 517–526, 2010.

[24] Y. Liu and T. Zhang, "On the Selection of Arithmetic Unit Structure in Voltage Overscaled Soft Digital Signal Processing," in *IEEE International Symposium on Low Power Electronics and Design*, 2007.

[25] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

Ku He Ku He received his B.E. and M.E. degrees in electrical engineering in 2004 and 2007 respectively, both from Tsinghua University. And he received his Ph.D. degree in 2012 from University of Texas at Austin. His research interests include low-power and robust circuit design.

Since 2012, he has joined Cirrus Logic, Inc. as a mixed-signal design engineer, whose tasks include designing high-resolution and low-power audio-band integrated circuits.

Andreas Gerstlauer Andreas Gerstlauer received the Dipl.-Ing. degree in electrical engineering from the University of Stuttgart, Germany, in 1997, and the M.S. and Ph.D. degrees in information and computer science from the University of California, Irvine (UCI), in 1998 and 2004, respectively.

Since 2008, he has been with the University of Texas at Austin, where he is currently an Assistant Professor in electrical and computer engineering. Prior to joining the University of Texas, he was an Assistant Researcher with the Center for Embedded Computer Systems, UCI, leading a research group to develop electronic system-level design tools.

Dr. Gerstlauer serves on the program committee of major conferences such as DAC, DATE and CODES+ISSS. His research interests include system-level design automation, system modeling, design languages and methodologies, and embedded hardware and software synthesis.

Michael Orshansky Michael Orshansky is an Associate Professor of Electrical and Computer Engineering at the University of Texas, Austin. He received his Ph.D. degree in Electrical Engineering and Computer Sciences from the University of California, Berkeley, in 2001. Prior to joining UT Austin, he was a Research Scientist and Lecturer with the Department of EECS at UC Berkeley. His research interests include design optimization for robustness and manufacturability, statistical timing analysis, and design in fabrics with extreme defect densities. He is the recipient of the National Science Foundation CAREER award for 2004 and ACM SIGDA Outstanding New Faculty Award in 2007. He received the 2004 IEEE Transactions on Semiconductor Manufacturing Best Paper Award, as well as Best Paper Awards at the Design Automation Conference 2005, International Symposium on Quality Electronic Design (ISQED) 2006, and International Conference on Computer-Aided Design (ICCAD) 2006. He is the author, with Sani Nassif and Duane Boning, of the book "Design for Manufacturability and Statistical Design: A Constructive Approach."