Energy and Quality-Aware Multimedia Signal Processing

Yunus Emre, *Student Member, IEEE* and Chaitali Chakrabarti, *Fellow, IEEE*

***Abstract*—This paper presents techniques to reduce energy with minimal degradation in system performance for multimedia signal processing algorithms. It first provides a survey of energy-saving techniques such as those based on voltage scaling, reducing number of computations and reducing dynamic range. While these techniques reduce energy, they also introduce errors that affect the performance quality. To compensate for these errors, techniques that exploit the algorithm characteristics are presented. Next, several hybrid energy-saving techniques that further reduce the energy consumption while keeping the performance degradation very low are presented. For instance, a combination of voltage scaling and dynamic range reduction is shown to achieve 85% energy saving in a low pass FIR filter for a fairly low noise level. A combination of computation reduction and dynamic reduction for Discrete Cosine Transform shows, on average, 33% to 46% reduction in energy consumption while incurring 0.5dB to 1.5dB loss in PSNR. Both of these techniques have very little overhead and achieve significant energy reduction with little quality degradation.**

***Index Terms*—low-power, voltage scaling, variable precision, error compensation, multimedia algorithms.**

I. Introduction

Portable multimedia devices have proliferated in the last two decades, and the number of applications supported by these devices has increased significantly. Each additional application comes at a cost of higher energy consumption and since most of these devices are battery powered, it is important that every effort be made to reduce it. The challenge is to minimize the energy cost while executing increasingly complex functionalities with minimal degradation in algorithm performance quality. Fortunately, many of the multimedia applications do not need 100% correctness during computation and energy saving transformations are favored as long as the output quality is mildly affected [1-2].

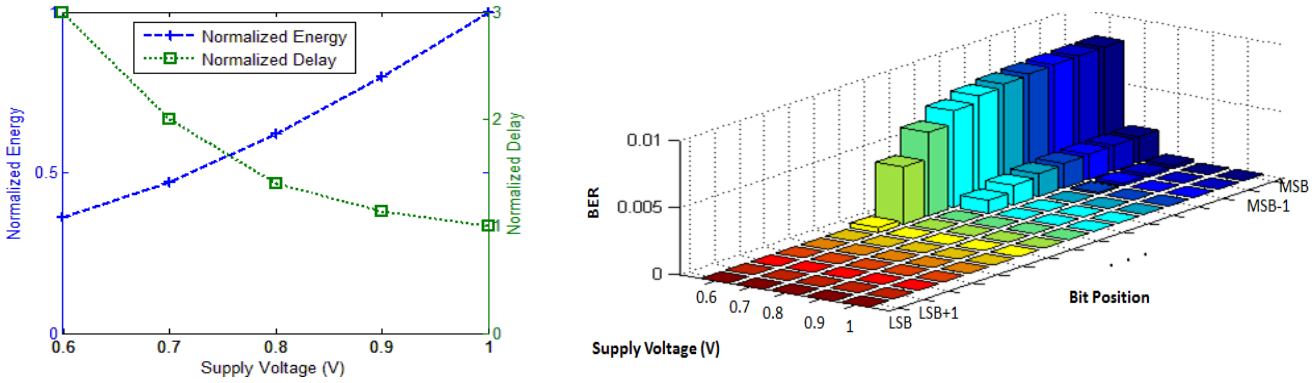
Three of the most effective techniques for reducing energy consumption are voltage scaling [3-14], reduction in number of computations [2, 15-20] and dynamic range adjustment [16, 18, 21-24]. While voltage scaling results in significant reduction in energy consumption due to the quadratic dependence between supply voltage and energy consumption, voltage over-scaling (VOS) can lead to failures. Techniques have been developed to mitigate the errors due to critical path violation in the computation unit and memory due to VOS. While circuit-level techniques [25-26] are quite effective, there are low overhead algorithm-level techniques that use the inherent redundancy and characteristics of the data to detect and correct errors that occurred during the computation.

Unlike general purpose computing, most multimedia applications can provide decent quality even with reduced number of computations as long as the significant computations are retained. The basic idea is that all components of the computation are not equally significant and so for systems with limited resources, the more important computations are done first and the less important computations are performed later or even eliminated. Such a methodology has been applied to many image and video processing algorithms such as filters, where multiplications with large coefficients have higher significance [2], Discrete Cosine Transform (DCT), where low frequency coefficients have higher significance [16], or Discrete Wavelet Transform (DWT) [20], where low subband coefficients have higher significance. This is also the basis of incremental processing where computations can be halted when decent quality is achieved [20].

Another popular energy-saving technique is dynamic range reduction in the datapath computation. Typically, low order bits are less important and so can be truncated to save energy. Such a methodology has been used in many multimedia applications such as filtering [16], DCT [22-23], FFT [5] etc. However, in some applications such as motion estimation [18], the less significant bit computations are more important since large values that are the result of computations with the more significant bits, are discarded. While truncation reduces energy consumption, it also introduces errors due to operation with a reduced dynamic range. Simple techniques to compensate for these errors help reduce energy consumption while mildly affecting the algorithm performance quality.

In this paper, we describe several energy-saving techniques that achieve minimum degradation in quality with low overhead. While some of these techniques are general, others have been geared to exploit the algorithmic features and result in superior performance both in terms of energy consumption and algorithm quality. The key contributions of this paper are as follows:

Y. Emre and C. Chakrabarti is with the School of Electrical, Computer and Energy Engineering at Arizona State University, AZ, USA. (yemre@asu.edu, chaitali@asu.edu)



(a) (b)

Figure 1:16-bit RCA under voltage scaling. a) Energy delay profile, b) Error distribution.

- We provide a survey of general as well as algorithm-specific techniques that trade-off energy with system performance for multimedia signal processing algorithms. Examples include FIR filtering, DCT, DWT etc. These techniques are based on voltage scaling in datapath and memory, reduction in the number of computations and reduction in the dynamic range.
- We propose hybrid schemes that use combination of these techniques to achieve even higher energy saving with smaller performance degradation. The performance overhead and energy savings of each scheme is quantified and analyzed.
  - We study the combination of voltage scaling and dynamic range reduction in the context of low pass FIR filter applications. The errors due to increase in critical path delay during voltage scaling are reduced by truncating the lower order bits which causes a reduction in the critical path. The noise that is introduced due to truncation is compensated by using an unbiased estimator. For a MAC based FIR filter, such a scheme achieves 85% energy saving for a fairly low noise level.
  - We study the combination of computation reduction and dynamic range reduction for DCT. We propose a scheme that chooses which DCT coefficients have to be deactivated and the number of bits to be truncated based on the quality metric, Q. We derive combinations of deactivation and truncation for different acceptable PSNR degradations across the whole range of Q. Simulation results show on average, 33% to 46% reduction in energy consumption while incurring 0.5dB to 1.5dB degradation in PSNR performance of JPEG.

The rest of the paper is organized as follows. A survey of low energy techniques is given in Section II followed by the hybrid schemes that combine different low energy techniques in Section III. Finally, Section IV concludes the paper.

II. Energy-Saving Techniques

In this section, we describe the three main techniques for reducing energy consumption, namely, voltage scaling (Section II.A), reducing number of computations (Section II.B) and reducing dynamic range in computation (Section II.C). We describe the errors introduced by each of the techniques and ways to compensate them to reduce algorithm-quality degradation.

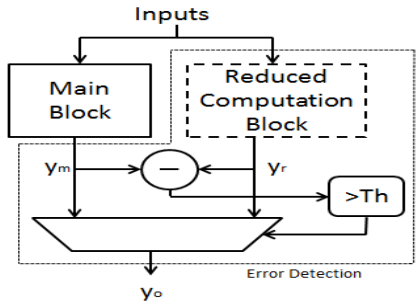


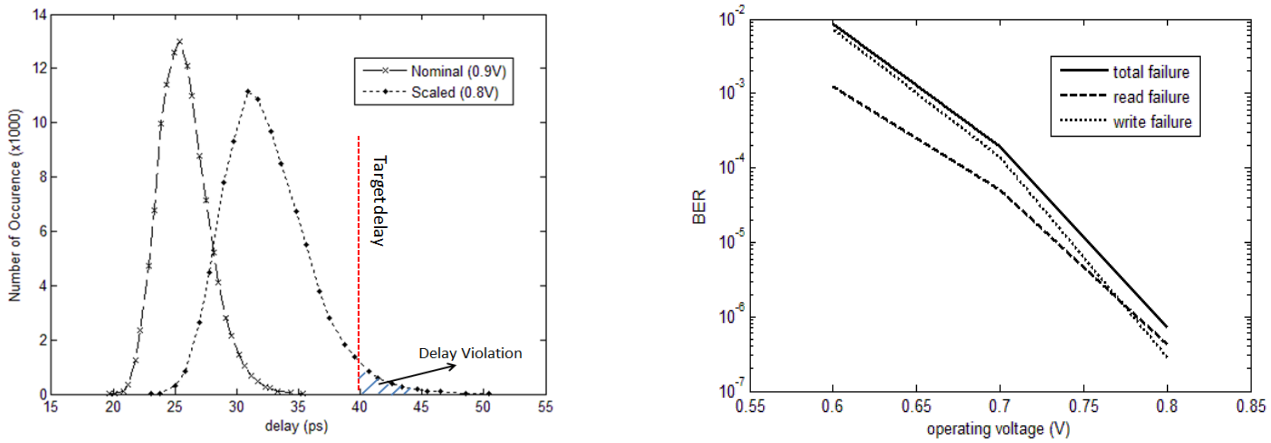
Figure 2: Block Diagram of Algorithm Noise Tolerant (ANT) Scheme

A. *Voltage Scaling*

One of the most effective techniques for energy reduction is voltage scaling. This is due to the quadratic dependence between energy consumption and supply voltage [3]. Figure 1a illustrates the normalized energy and delay plots of a 16-bit ripple carry adder (RCA) as a function of supply voltage. This was obtained by our in-house simulator based on modelSim with 45nm PTM models [27]; the normalization is with respect to 1V nominal voltage operation. Voltage overscaling (VOS) refers to scaling the voltage beyond the value imposed by the critical delay of the circuitry. This may result in timing violations in the data-path, resulting in erroneous operation. Figure 1b illustrates the error distribution of the 16-bit RCA under voltage scaling [8]. Note that most of the errors reside in the most significant bits, which can result in significant performance degradation.

A.1. Compensating Datapath Errors

To mitigate the errors due to critical path violation of the computation unit under voltage scaling, algorithm noise tolerance (ANT) has been used in [4-7]. Figure 2 illustrates the general block diagram of the ANT scheme which consists of



(a) (b)

Figure 3: a) Delay distribution of a single SRAM read operation; b) Read, write and total failure probability of SRAM for different voltage levels in 32nm

the main block and the reduced computation block. The main block implements the original computation at full precision; thus its output (y_m) is prone to critical path violation under voltage scaling. The reduced computation block is designed to generate a statistical replica (y_r) of the original result with a shorter critical path. These two outputs are compared to detect any errors that may have occurred in the main block. Since VOS results in large errors in magnitude, the system chooses y_m if the difference is smaller than the predetermined threshold (Thr) and y_r otherwise.

In an ANT based system, the reduced computation block needs to provide good approximation of the original output, have low complexity circuitry to minimize overall overhead, and have shorter critical path to ensure error-free operation. Reduction in computation is achieved by using reduced precision replica [4-5], subsampling [6], and prediction based error correction [7]. ANT-based systems have been applied to multimedia applications such as FIR low pass filter [4-5], FFT [5], and motion estimation [6]. In [4], correlation of the FIR low pass filter outputs is used to correct errors if any. To minimize the overhead, a very simple low pass filter is employed that computes the estimates of the main block. In [5], the reduced computation block is based on 4-bit MSB implementation of FFT while the main block operates on 8-bit data. In [6], a subsampled version of the original motion estimation block is used in the reduced computation block. All these methods achieve 20% to 40% energy reduction while incurring small performance degradation.

Recently, an algorithmic-specific technique has been proposed in [8] that mitigates datapath errors during the computation of 2D-DCT and quantization in JPEG. The technique exploits encoded JPEG data features to detect the VOS induced errors. The features are based on facts such as two adjacent AC coefficients after zig-zag scan have similar values, coefficients corresponding to higher frequencies generally have smaller values and the number of sign extension bits is determined in the quantization step. The technique achieves high performance with small circuit overhead. Simulation results show that the proposed technique has a PSNR performance degradation of around 1.5dB compared to the error-free case, and 4dB improvement compared to the no correction case at compression rate of 0.75 bpp when $BER = 10^{-4}$. The overhead of this technique is quite small; it requires three simple units, namely, a 3-bit majority voter, a 8-bit coefficient comparator and a 8-bit 2 input average calculator.

A.2. Compensating Memory Errors

SRAM failure analysis under voltage scaling have been investigated by several researchers [9-11, 28-31]. In [28], statistical models of random dopant fluctuations (RDF) are used to determine read, write failure and access time variations. In [29], read and write noise margins of 6T SRAM cells are used to calculate the memory reliability. Figure 3a illustrates the distribution of read access times at nominal and scaled voltage levels for a 32nm SRAM cell under 40mV RDF and 5% channel length variation. From the figure, we see that as the voltage scales down, the tail section of the access time become heavier, which manifests itself in access time errors at scaled voltages. Figure 3b illustrates the read, write and total failure rates as the voltage scales from 0.8V to 0.6V [32]. At nominal voltage of 0.9V, the BER is estimated to be 10^{-10}. At lower voltages, the BERs are very high. For instance, at 0.7V, the BER is 10^{-4} and at 0.6V, it climbs to 10^{-2}. Such high error rates were also reported in [10, 29].

Several circuit, system and architecture-level techniques have been proposed to mitigate and/or compensate for memory failures. At the circuit level, different SRAM structures such as 8T and 10T have been proposed [10, 31]. In 8T and 10T structures, data path for read and write operation are separated to increase the robustness of the operations. However, the additional circuitry increases leakage power and circuit area by approximately 20% to 30%. The method in [10] stores the MSBs in a memory bank with 8T SRAM cells and the least significant bits (LSB) in memory banks with 6T cells. It uses the fact that 8T SRAM cells are more robust then basic 6T SRAM cells at scaled voltages. Such a scheme achieves approximately 40% power reduction while having 15% increase in overall circuitry area. The method in [11] operates

the memory banks that store MSBs at a different voltage level than the ones that store LSBs. This is shown to achieve 45% power reduction with 10% degradation in image quality for a regular pixel based image-storage system.

Many techniques make use of error control coding (ECC) [8, 12, 32-34]. In [12], orthogonal latin square codes are used to trade-off cache size with correction capability. Extended Hamming codes which provide single error correction, double error detection (SECDED) have been used for several years to combat failures in memory systems [32-34]. Their simple structure makes them appealing for applications that require low latency and power consumption. The memory area overhead of the stronger codes is very large, thus using unequal error protection (UEP) that combines strong and weak codes, is a better option. The main idea of UEP is to provide superior protection to the more important bits and thus enable the area overhead to be reduced without sacrificing performance [32]. For instance, in JPEG2000, the higher subband DWT outputs are more important and so should be protected better with stronger codes. In [32], it has been shown that compared to single ECC, UEP based on different SECDED codes has 35% lower MSE for the same overhead.

Algorithm-specific techniques have also been developed for memory intensive multimedia applications in [13, 32, 35]. These techniques mitigate system degradation due to memory failures using additional features that are intrinsic to the algorithm. In [13], binarization and second derivative of the image are used to detect error locations in different DWT sub-bands in JPEG2000. These are then corrected in an iterative fashion by flipping one bit at a time starting from the MSB. In [35], application-aware methods have been proposed to reduce power consumption of memories in video coding systems under VOS while maintaining the performance using simple filters after processing. Algorithm-specific techniques that exploit the characteristics of the DWT coefficients have been proposed in [32] for JPEG2000. These techniques identify and correct errors by exploiting the fact that DWT outputs at high subbands typically consist of smaller values and thus contain small number of non-zero bits in MSB planes. Also, there is a similarity between magnitudes of neighboring coefficients. Based on these features, an error is flagged when isolated non-zero MSBs are detected at high frequency bands of the DWT output. These techniques achieve performance results close to error-free curves (only 1dB degradation in PSNR) at 0.75bpp when BER=10^{-3}. Even for very a high bit error rate such as BER=10^{-2}, the algorithm-specific scheme can achieve 7.9dB performance improvement compared to the no correction case, 4dB to 5dB improvement compared to the (39, 32) Hamming ECC scheme and only 2.8dB degradation compared to the no-error case. The overhead of these techniques is very small; the additional circuits include a 9-bit counter, 35-bit all zero detector and a 4-bit comparator.

B. Reducing Number of Computations

The number of computations can be reduced by choosing a smarter algorithm with a lower complexity. Examples of this include Fast Fourier Transform implementation of the Discrete Fourier Transform, differential tree search based vector quantization [3], etc. However, most of the time reduction in the number of computations comes with some performance hit. For instance, in block matching that is used in motion estimation, heuristic algorithms such as three step search and diamond search have lower complexity but sub-optimal performance. These search algorithms are used when the performance requirements are not that stringent and the available energy is low [19]. For each search algorithm, sub-sampling can be used to further reduce the number of computations [6, 17, 18]. If 1/s is the subsampling ratio, then these schemes reduce the number of absolute difference computations in motion estimation by 1/s and result in significant energy reduction. However there is a performance cost, for instance, s=2 increases the compression rate by 6.5% and reduces the PSNR by approximately 0.4dB [18].

An effective way of reducing the number of computations with minimal degradation in system performance is by exploiting the fact that different portions of the computation have different levels of significance on the overall system quality. One of the earliest works in this area was done for least mean square (LMS) type adaptive filters in [36], where the filter length was determined based on the energy consumption and system performance requirements. In DCT implementation, for instance, most of the image energy of the DCT resides in the low frequency coefficients and higher frequency coefficients can be sacrificed when good enough quality is achieved [2, 16]. Similarly, in FIR filtering, larger filter taps contribute more to system performance, and so sorting the impulse response of the filter taps in decreasing order of magnitude and computing on larger coefficients first helps achieve energy saving with reduced overall quality degradation [16]. Other examples include multilevel DWT where the coefficients are computed incrementally one bit plane at a time till the desired quality is achieved [20], and salient point detection where the detection result is refined as the image precision improves [37].

C. Reducing Dynamic Range in Computation

Reducing the datapath precision to lower power consumption is a popular technique in signal processing systems. Typically, high order bits contain most of the information while low order bits capture the details of the application. Figure 4 illustrates the savings in energy consumption of a 16-bit RCA for different bit widths in 45nm technology. Since RCA has a regular structure, the energy reduction is proportional to the bit-width of the adder. For instance, at nominal voltage, we observe 24% reduction in energy consumption of the adder when 12-bits are used instead of 16-bits.

One drawback of reduced precision arithmetic is that it introduces truncation errors. Figure 4 also plots truncation noise defined as the magnitude of the difference between the output obtained with full precision data and the output obtained with truncated data scaled by the full precision

output. From Figure 4, we see that while truncation noise increases logarithmically, energy saving of the adder increases linearly with increase in number of truncated bits. Low order bit truncation can easily be applied to other multimedia applications such as filtering, DCT; however, one of the main challenges is to compensate for the quality degradation caused by reduced precision.

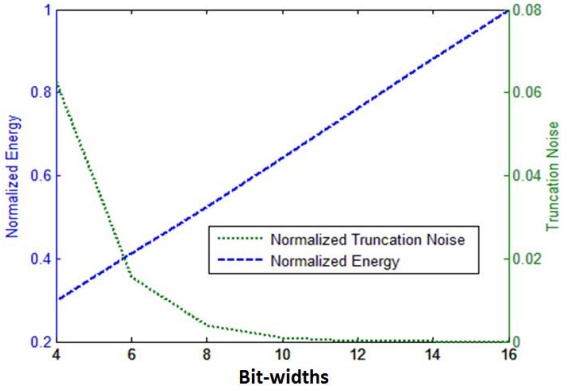


Figure 4: Normalized energy and truncation noise as a function of bit-width

Additional energy saving can be obtained by approximating the computations in the datapath components [38-43]. Adders and multipliers that trade-off accuracy for lower power consumption by reducing the carry chain have been proposed in [38-40]. For instance, the modified Kogge-stone adder in [38] operates on a shorter critical path but the errors are not as significant since the probability of having timing violations with shorter carry chain is not very high. The RCA based error-tolerant adder proposed in [39] partitions the carry chain into variable width segments in which MSB side has longer segments than LSB side to reduce the error. Multiplier architectures in [40] truncate the partial product generation at the LSB end resulting in small truncation noise while achieving significant energy savings.

In addition, the building blocks of the adders and multipliers can also be approximated by selectively removing some minterms of their Boolean functions [41-43]. For instance, [42] describes a 2x2 under-designed, inaccurate multiplier which is used to implement a larger multiplier for image processing applications. Based on the system requirements, a correction term is introduced to reduce the degradation in algorithmic performance. A more general scheme is proposed in [43] to reduce the area of combinational logic for a given error rate threshold. During the synthesis phase, the number of literals used in the logic function is reduced by complementing the minterms of the original function.

C.1. Low Order Bit Truncation

Bit truncation methods that remove low order bits have been very effective for motion estimation [17, 18, 24]. In [24], instead of using 8 bits, only 4 or 5 of the higher order bits are kept to reduce the activity in less important regions. In [18], the performance degradation and increase in compressed data rate have been studied for low order bit truncation in motion estimation used in H.264. Figure 5 illustrates the average degradation over several video sequences for low order bit truncation ranging from 1-bit to 4-bits for diamond (DS) and three step search (TSS) strategies [18]. Here the performance metrics are ΔCR which represents the change in compression rate and $\Delta PSNR$ which represents the change in PSNR; lower ΔCR and $\Delta PSNR$ corresponds to better quality. Since motion estimation is based on subtraction of the pixel values, the expected performance degradation is not very high. This is because subtraction is more tolerant to truncation noise than addition or multiplication operations. In algorithms whose building blocks are multiplications and additions such as DCT and FIR filter, the truncation error has to be compensated to maintain good image quality. Next, we describe the proposed technique for compensating truncation error.

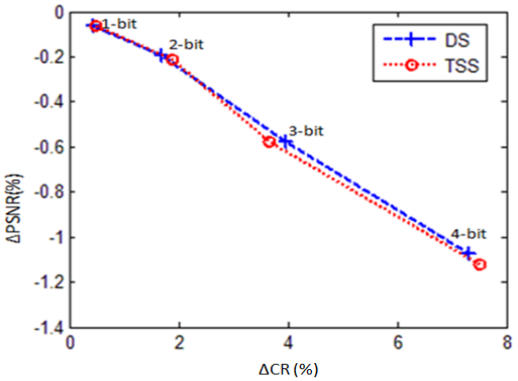


Figure 5: Average performance loss due to bit truncation using DS and TSS

Truncation Errors: Analysis and Compensation

First, we investigate the effect of bit truncation on simple arithmetic operations such as addition, subtraction and multiplication. We describe the error characteristics for operations on unsigned numbers; however the procedure can easily be extended to operations on two's complement and signed numbers. Next, we describe a method to reduce the effect of truncation based errors on system quality.

The output of a DSP system after LSB truncation at time instant *k* can be expressed as:

$$y_e[k] = y_o[k] + v[k]$$

where $y_o[k]$ is the truncation-free output and $v[k]$ is the truncation induced error (noise) which is a random variable with mean μ_N and variance σ_N^2. The noise power can be represented by the mean square error (MSE) defined as $\mu_N^2 + \sigma_N^2$. In order to reduce the noise power, we propose a method that estimates the mean value of the truncation error during the pre-computation stage and compensates for it. We refer to this method as μ_N compensation. The overhead of this method is very small. Moreover the noise power after μ_N-compensation does not depend on μ_N anymore and is only a function of the variance of the truncation error.

Let us consider a system whose inputs are originally represented with $M+1$ bits, $x(M:0)$. When L bit truncation is employed, where $L \leq M$, the input becomes $x(M:L)$. Assuming uniformly distributed input signals, we can express

Q_x, the truncation error for the input signal x, as:

$$Q_x(L-1:0) = x(M:0) - x(M:L) = \sum_{i=0}^{L-1} 2^i b_i$$

where b_i is an independent, uniform random variable with two discrete values: 0 and 1. The expected value and variance of Q_x are given by,

$$E[Q_x(L-1:0)] = \mu_q(L-1:0) = \frac{1}{2}(2^L - 1) \quad (1)$$

$$var[Q_x(L-1:0)] = \sigma_q^2(L-1:0) = \sum_{i=0}^{L-1} 4^i \sigma_{b_i}^2 = \frac{4^L-1}{12} \quad (2)$$

where $\mu_q(L-1:0)$ and $\sigma_q^2(L-1:0)$ are mean and variance of $Q_x(L-1:0)$ and $\sigma_{b_i}^2$ is the variance of b_i.

Using equation (1-2), we can compute the expected value and variance of the truncation error (Q_{add}) of an adder with inputs x and y. Both inputs are independent and in both cases the lower L bits (out of M+1 bits) have been truncated.

$$\begin{aligned} E[Q_{add}(L-1:0)] &= E[x(M:0) + y(M:0) - x(M:L) - y(M:L)] \\ &= 2\mu_q(L-1:0) \\ var[Q_{add}(L-1:0)] &= 2\sigma_q^2(L-1:0) \end{aligned}$$

Using similar analysis, we compute the expected value and variance of subtraction and multiplication. Details of the calculation for multiplication are given in the Appendix.

$$\begin{aligned} E[Q_{sub}(L-1:0)] &= 0 \\ var[Q_{sub}(L-1:0)] &= 2\sigma_q^2(L-1:0) \end{aligned}$$

$$\begin{aligned} E[Q_{mul}(L-1:0)] &= 2\mu_q(M:0)\mu_q(L-1:0) - \mu_q(L-1:0)^2 \\ var[Q_{mul}(L-1:0)] &= 2[\mu_q(L-1:0)^2\sigma_q^2(M:L) \\ &\quad +\mu_q(M:L)^2\sigma_q^2(L-1:0) + \sigma_q^2(L-1:0)\,\sigma_q^2(M:L)] \\ &\quad +2\mu_q(L-1:0)^2\sigma_q^2(L-1:0) + \sigma_q^2(L-1:0)\sigma_q^2(L-1:0) \\ &\quad +2\mu_q(L-1:0)\mu_q(M:L)\sigma_q^2(L-1:0) \end{aligned}$$

Figure 6 illustrates how the noise power (MSE) of 16-bit multiplication of unsigned numbers can be reduced with μ compensation. We see that the analytical results and simulated results match very closely. Moreover μ plays an important role in determining the noise power and compensating for μ helps reduce the MSE by >2X.

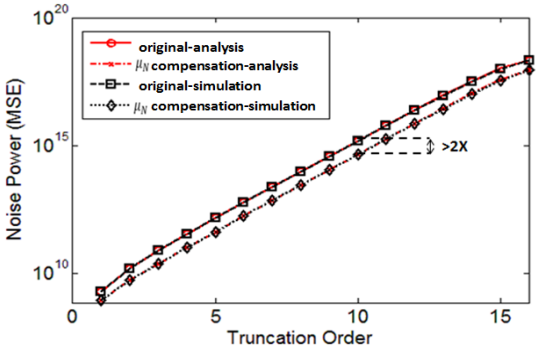


Figure 6: Noise power of Q_{mul} for 16-bit multiplication with and without μ_N-compensation

Since noise power is proportional to $\mu_N^2 + \sigma_N^2$, the proposed method helps in reducing the noise power for computations such as additions and multiplications. It does not help in the case of subtractions since the μ_N of subtraction is 0. Furthermore, we see that, noise power of an 'L' bit truncation with compensation and 'L-1' bit truncation without compensation are comparable. However since the overhead of μ_N compensation is very small, a system with larger number of truncation bits has larger energy savings as will be illustrated in Section III.C. Next we illustrate the use of the μ_N compensation method to compensate for errors in DCT and FIR filter computation.

Example 1 - Discrete Cosine Transform (DCT): 2-D DCT is typically implemented using 1-D DCTs along rows (columns) followed by 1-D DCT along columns (rows) as illustrated in Figure 7.

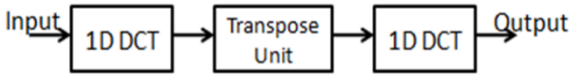


Figure 7: 2D DCT architecture using 1-D DCTs

1-D DCT transform of size 8 that is used in JPEG can be expressed as follows:

$$W_i = \frac{c_i}{2}\sum_{k=0}^{7} x_k \cos\frac{(2k+1)k\pi}{16} \qquad c_i = \begin{cases} \frac{1}{\sqrt{2}} & i = 0 \\ 1 & i = 1, \ldots, 7 \end{cases}$$

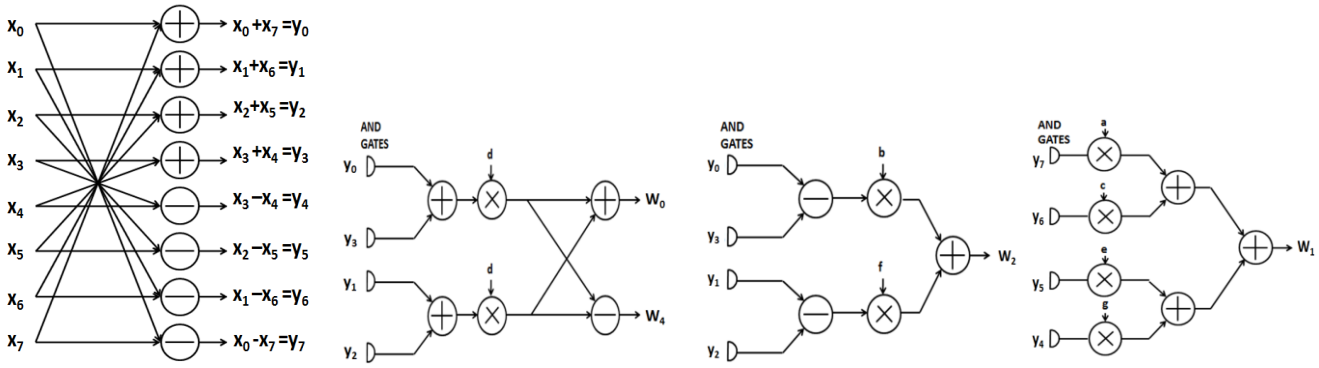
where x_k's are input pixels in row or column order and w_i's are the corresponding outputs. Typically 8-point DCT is computed along rows and the coefficients transposed so that data for the 8-point DCT along columns can be obtained efficiently. The properties of the coefficient matrix are used to reduce the number of multiplications. Below is one such method of implementing the odd and even coefficients.

$$\begin{bmatrix} W_0 \\ W_2 \\ W_4 \\ W_6 \end{bmatrix} = \begin{bmatrix} d & d & d & d \\ b & f & -f & -b \\ d & -d & -d & d \\ f & -b & b & -f \end{bmatrix} \begin{bmatrix} x_0 + x_7 \\ x_1 + x_6 \\ x_2 + x_5 \\ x_3 + x_4 \end{bmatrix}$$

$$\begin{bmatrix} W_1 \\ W_3 \\ W_5 \\ W_7 \end{bmatrix} = \begin{bmatrix} a & c & e & g \\ c & -g & -a & -e \\ e & -a & g & c \\ g & -e & c & -a \end{bmatrix} \begin{bmatrix} x_0 - x_7 \\ x_1 - x_6 \\ x_2 - x_5 \\ x_3 - x_4 \end{bmatrix}$$

where $a = \frac{1}{2}\cos\left(\frac{\pi}{16}\right), b = \frac{1}{2}\cos\left(\frac{2\pi}{16}\right), c = \frac{1}{2}\cos\left(\frac{3\pi}{16}\right), d = \frac{1}{2}\cos\left(\frac{4\pi}{16}\right), e = \frac{1}{2}\cos\left(\frac{5\pi}{16}\right), f = \frac{1}{2}\cos\left(\frac{6\pi}{16}\right), g = \frac{1}{2}\cos\left(\frac{7\pi}{16}\right)$.

Figure 8 describes the architecture to compute 4 DCT coefficients (W_0, W_1, W_2 and W_4) of the 8-point DCT used in JPEG. After pairwise subtraction and addition of the pixels, we obtain y_0 to y_7, where $y_0 = x_0 + x_7$, $y_1 = x_1 + x_6$, …, and $y_7 = x_3 - x_4$. For W_0 and W_4, common sub-expression elimination is used to obtain results with small number of computation units as illustrated in Figure 8b. Implementation of W_2 is illustrated in Figure 8c; a variant of which is used for W_6. Figure 8d shows the computation structure used to find W_1. The odd coefficients (W_3, W_5 and W_7) are computed using units that are similar to the unit for W_1.



(a) (b) (c) (d)

Figure 8: Computation of 1-D DCT coefficients. a) First stage butterfly; b) W_0 and W_4 computation unit; c) W_2 unit; d) W_1 unit

We calculate the truncation noise (TN) for the DCT outputs for a 14 bit fixed point implementation of DCT, where 12 bits represents the integer part and last 2 bits represent the fractional part of the computation. The expected errors due to truncation in W_0 and W_1 can be expressed as follows. To simplify our analysis, we assume that all Y values are uncorrelated and so the expected value for L bit truncation is $\frac{2^L-1}{8}$. Since $W_0 = d * (y_0 + y_1 + y_2 + y_3)$, the expected truncation error for W_0 is given by $TN_{W_0} = E\left[d * \left(y_0(L-1:0) + \cdots + y_3(L-1:0)\right)\right] = \left\lfloor \frac{d(2^L-1)}{2} \right\rfloor$. Similarly expected value of the truncation errors for W_1 is given by $\left\lfloor (a+c+e+g)\ \frac{2^L-1}{8} \right\rfloor$, and that of W_2 is given by $(b+f-b-f) \times E[y] = 0$. The expected value of truncation noise for W_4 and W_6 are also zero.

The expected truncation noise values are used as unbiased estimators to compensate the errors. Instead of compensating for errors of all the outputs, we only compensate for errors in the computation of W_0 and W_1. The motivation for this is that these coefficients are the most important ones and the corresponding estimation errors are the largest. Also this keeps the complexity of the overhead circuitry small. Figure 9 illustrates the compensation mechanism for W_1 computation. The overhead of this scheme is the 14-bit adder at the output as well as the AND gates to disable a selective set of input bits. The area and power overhead due to extra processing elements is around 2% of the overall DCT implementation.

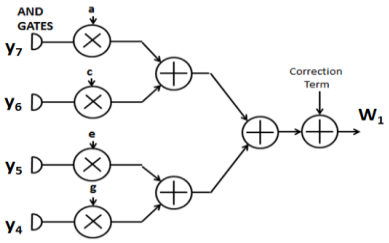


Figure 9: Modified DCT computation of W_1

Figure 10 illustrates the performance improvement with the use of unbiased estimators for W_0 and W_1 when low order bits are truncated for DCT computation of the Baboon image. For 1 bpp compression rate, 4-bit truncation causes a degradation of 1.3dB which is reduced to 0.6dB with compensation. For the same 1 bpp compression rate, when 6 bits are truncated, the performance improvement is approximately 1.2dB compared to the system without compensation. Thus as the truncation level increases, we observe higher performance improvements in systems that use compensation.

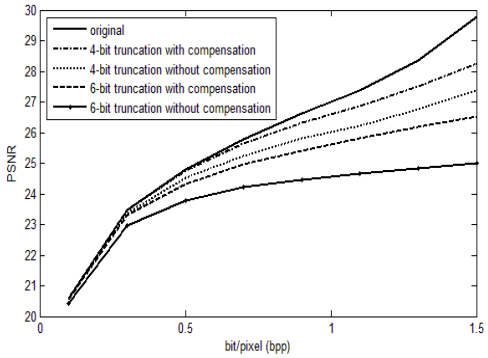


Figure 10: Performance comparison between uncompensated and compensated bit truncation for DCT computation of Baboon image

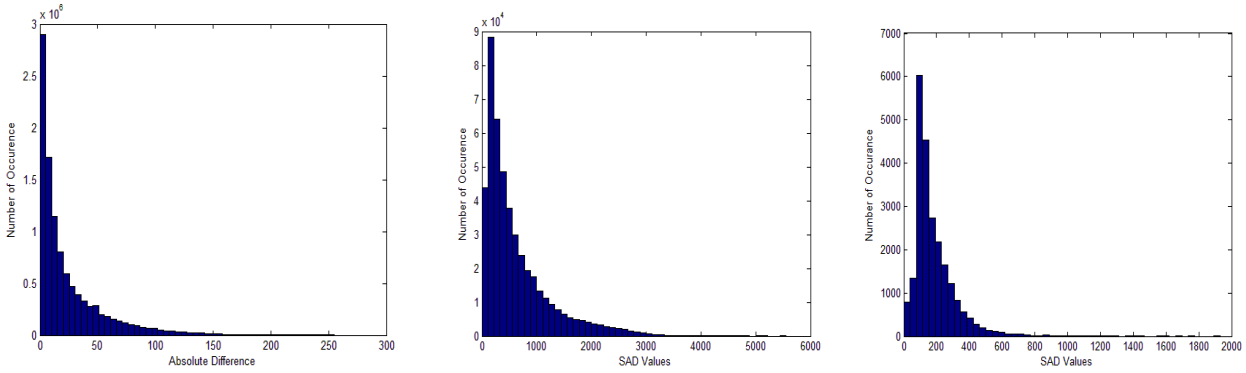
***Example 2*:** Consider a FIR low pass filter (LPF) using unsigned inputs and coefficients, which is typical in many multimedia algorithms. The output y(n) of an N-tap filter with M+1-bit precision is given by

$$y(n, M:0) = \sum_{k=0}^{N-1} h(k, M:0)x(n-k, M:0)$$

where $h(k, M:0)$ is the k'th coefficient of the filter and x(n-k,M:0) represents the input value at time n-k. Such a computation can be implemented efficiently using MAC based architectures. We can calculate the unbiased estimator for L-bit truncation assuming coefficients are less than one as:

$$E[Q_x] = E[\sum_{k=0}^{N-1} h(k, M:0)x(n-k, M:0) - h(k, M:L)x(n-k, M:L)] \quad (3)$$

When filter coefficients are known, the estimator given in equation (3) reduces to:



(a) (b) (c)

Figure 12: Histogram plots of a) AD values, b) SAD values of all blocks, and c) selected SAD values for the Football sequence

$E[x(L-1:0)]\sum_{k=0}^{N-1}h(k,M:0) + E[x(M:L)]\sum_{k=0}^{N-1}h(k,L-1:0)$

$$= \frac{(2^L-1)}{2}DC_{gain} + \frac{(2^{M+1}-2^L)}{2}\sum_{k=0}^{N-1}h(k,L-1:0)$$

where DC_{gain} represents the sum of filter tap coefficients for LPF given by $DC_{gain} = \sum_{k=0}^{N-1}h(k,M:0)$. As an example, for a 3x3 Gaussian Filter ($\sigma = 1$) with M=7 and L=2 the unbiased estimator value is 3; this value increases to 15 when L=4.

Figure 11 illustrates the block diagram of the proposed MAC based architecture for LPF. Filter coefficients and input data are truncated using an array of AND gates before the multiplication; thus only high order bits become active during computation. After N cycles of MAC computation, the correction factor is applied to reduce errors due to truncation. The performance results of this filter are given in Section III.

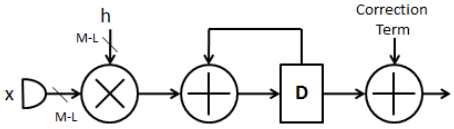


Figure 11: MAC implementation of a low pass filter

C.2. High Order Clipped Computation

It is not always the case that the low order bits are less significant in computation and so can be dropped. In sum of absolute difference (SAD) computation used in motion estimation, for instance, the high order bits can be dropped. The proposed scheme in [18] uses the statistics of absolute difference (AD) and SAD computations to reduce the dynamic range and approximate the computations. Specifically, it exploits the fact that most of the AD values are small due to locality of current and reference blocks, and that most of the large AD values are for blocks that are likely not to be selected, and thus these values can be approximated. Figure 12 illustrates the distribution of the AD values, SAD values and selected SAD values for the Football video sequence. From the distributions, we see that the dynamic range of the selected SAD values is significantly lower than the dynamic range when all SAD values are taken into consideration. Thus during SAD computation, it is not necessary to operate on the MSBs during SAD computation with much care. The scheme in [18] detects large AD values using special logic and the corresponding SAD values are updated with a correction factor. The resulting architecture has a lower critical path delay compared to the baseline architecture and significantly lower energy consumption. It achieves 37.5% energy reduction at nominal voltage and 68% reduction for iso-throughput while incurring 1.8% increase in compressed data size and approximately 1.3 dB reduction in PSNR.

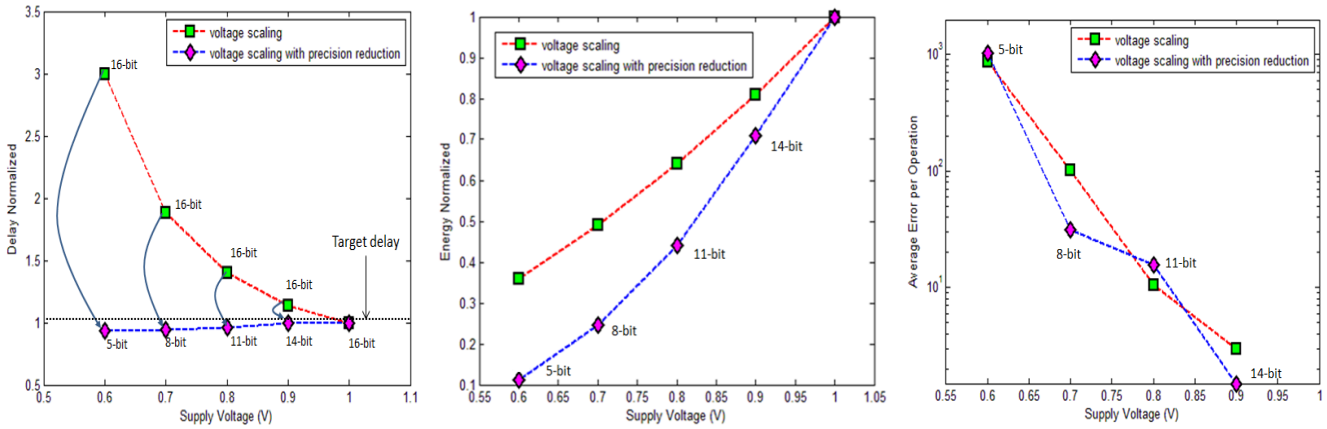
III. HYBRID CONFIGURATIONS

Each technique described in Section II achieves energy saving by itself; however their combinations achieve even higher energy saving as will be demonstrated in this section.

A. Combining Computation Reduction and Voltage Scaling:

Several significance driven techniques where the significant components have shorter delay and the less significant components have longer delay, have been proposed in [44-47]. These techniques are very effective in reducing the energy consumption without affecting the quality too much. At nominal voltage, all computations ensure no-violation in the critical path while at scaled voltage levels those which have higher critical path delay than the operating frequency are disabled. For instance, selective deactivation of DCT coefficient based on the operating voltage has been proposed in [44]. Since low frequency DCT coefficients contain most of the input image energy, they are significant and implemented with shortest critical path. It has been shown that 41% to 90% power saving is possible compared to baseline scheme with up to 10dB degradation in PSNR. A similar approach is applied in [45] for color interpolation where only less important computations are affected by voltage scaling and process variation. Such a scheme achieves 40% power savings with 5dB PSNR degradation.

These significance driven techniques have also been applied to support vector machines that are widely used in data-mining



(a) (b) (c)

Figure 13: 16-bit RCA. a) Delay, b) Energy, c) Average Error per Operation distribution with and without precision reduction under voltage scaling

applications [47]. Here the number of support vectors and features per vector are traded-off and voltage over-scaling is used at the circuit level to minimize the energy for a given quality of service. A more general method has been proposed in [48] where the behavior and cost of the computation components are modeled based on their importance on system performance. The supply voltage levels for a specific part of the circuitry are determined based on bit significance (profit) and energy cost (investment).

B. Combining Voltage Scaling and Dynamic Range Reduction

Voltage scaling and dynamic range reduction are two complementary energy reduction techniques. While voltage scaling reduces the energy consumption, it increases the delay of the computation unit and can cause timing errors. However, if reduced precision operation is acceptable, the critical path of the computation is lower and timing errors due to voltage scaling can be avoided. We illustrate this with the help of a 16 bit adder example and then show the effectiveness of this method in achieving energy reduction with minimal quality degradation for a low pass FIR filter.

Consider a simple 16-bit RCA implemented using modelSim with 45nm PTM model and simulated for uniformly distributed inputs. Figure 13a illustrates the change in critical path delay under voltage scaling; the target delay of the adder at nominal voltage and full precision is illustrated with a dashed line parallel to x-axis. As expected, the critical path delay of the 16-bit adder increases rapidly with voltage scaling. For instance, at 0.8V the increase in critical path delay is approximately 45% of the target delay.

The increase in critical path delay is reduced using lower precision arithmetic unit that has shorter critical path. For instance, voltage scaling induced errors due to critical path violation when operating the 16-bit adder at 0.8V is prevented by truncating 5 low order bits and operating only on the 11 MSBs. Similarly, critical path violation at 0.7V is prevented by operating on the 8 MSBs. Figure 13b shows the difference in energy saving between a scheme where only voltage scaling

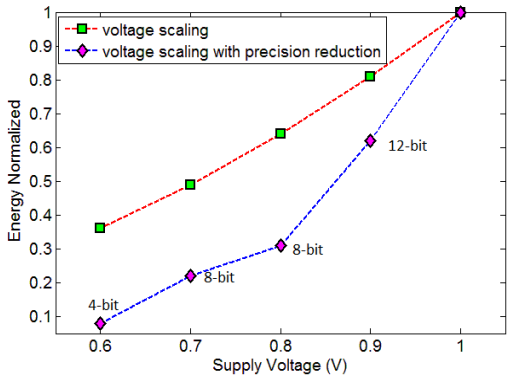
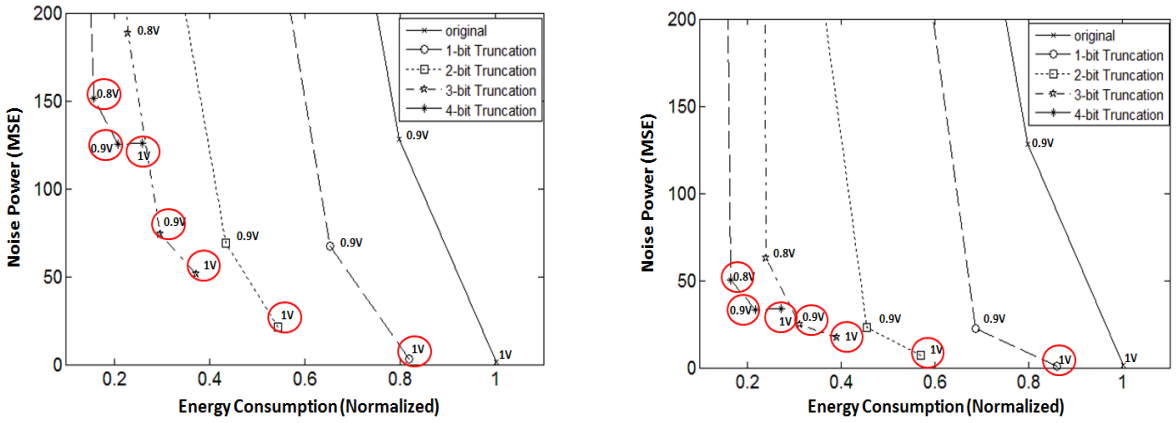


Figure 14: 16-bit CLA Energy distribution with and without precision reduction under voltage scaling for comparable average error per operation

is used and a scheme where a combination of voltage scaling and reduced precision is used. At 0.6V, use of only voltage scaling reduces the energy consumption by 63% while the combination reduces it by 89%. Propagation delay of an RCA is proportional to its width, thus, the delay of an N bit adder with L bit LOB truncation can be approximated as $\frac{N-L}{N}T_d$ where T_d is the propagation delay of N-bit RCA. Thus, the bitwidth of the RCA which prevents the VOS induced errors can be estimated easily.

Next, we analyze the average error induced by voltage scaling and the combined technique. At nominal voltages both systems operate at full-precision, and so the average error is zero. At scaled voltage levels, both systems have comparable average error per operation as shown in Figure 13c. For instance, at 0.8V, the 11-bit adder and the 16 bit adder have the same error/operation but the 11-bit adder has 20% lower energy (Figure 13b). The effect of VOS on adders has been formulated in [8, 49] which use internal architecture of the adders to estimate the noise power. Furthermore, the truncation noise can be lowered using the compensation technique described in Section II. Even without compensation, the combined technique achieves much higher energy saving compared to using only voltage scaling for a comparable error per operation.



(a) (b)

Figure 15: Noise Power vs Energy Consumption of 3x3 Gaussian filter on Lena image under voltage scaling with a) precision reduction, b) precision reduction and compensation

We see similar trends in delay, energy and error performance for more complex adders such as carry-look ahead adder (CLA). Figure 14 shows normalized energy as a function of the supply voltage for a 16-bit CLA. We see that CLA supports more aggressive truncation, for instance, when operated at 0.8V, it uses only 8 bits (out of 16 bits) and thus achieves higher energy savings. However, the average error per operation for the CLA is typically larger compared to RCA for the same voltage level and RCA tends to have better energy performance for the same error level. This is in agreement with the results presented in [49].

Next, we present the results of this procedure on real image data. Consider processing the Lena image with a 3x3 Gaussian filter (σ=1) using a MAC based architecture with 8-bit resolution. The multiplier is implemented using a carry save adder tree and the final stage is implemented using RCA. Also, both the inputs and the filter coefficients are truncated with the same order. Figure 15a shows the mean squared error noise power (VOS induced + truncation) vs. normalized energy consumption for various levels of low order bit truncation without compensation. Each point in the curve corresponds to a specific supply voltage level. Noise power is calculated using the mean square error (MSE) between the LPF results obtained with voltage scaling and those obtained with nominal voltage operation. (Note that MSE can be converted to PSNR using $20log(peak\ signal\ power/\sqrt{MSE}$; however we choose to use MSE here since it provides greater insight into the error performance). From this figure, we see that full precision LPF (original) shows a large increase in noise level when the voltage is scaled to 0.9V with only about 20% energy saving. On the other hand, 2-bit truncation operating at 0.9V has lower noise power and 45% energy saving. Thus, dynamic precision adjustment with voltage scaling achieves considerable better performance compared to when only voltage scaling is used.

Next, we study the effect of truncation noise compensation. Figure 15b illustrates the performance of the LPF when the estimator described in Section II.C.1 is applied.

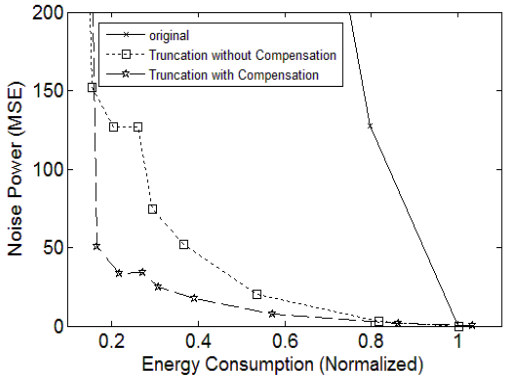


Figure 16: Performance comparison of 3x3 Gaussian filter on Lena image; original and reduced precision filter with and without compensation

For 4-bit truncation operating at 0.9V, the noise power reduces by 66% when compensation unit is used. The overhead is very small, since the compensation unit is activated only 1/N of the time where N is the number of the filter taps. At full precision, we have approximately 5% overhead compared to original MAC unit because of the final adder illustrated in Figure 11.

Finally, Figure 16 shows the pareto-optimal curves for voltage scaling in combination with truncation with and without compensation. These curves are generated by connecting the best configurations shown in circles in Figures 15a and 15b. We see that the combination scheme always achieves better performance compared to sole voltage scaling at all levels. Furthermore, truncation with compensation achieves higher energy saving for the same noise power. For instance, at MSE=35 (which is approximately PSNR=32.7), FIR using compensation achieves 16% extra energy saving compared to FIR with no compensation.

We repeat the analysis for three different 3x3 Gaussian filters ($\sigma = 0.5, \sigma = 1$ and $\sigma = 1.5$) and for two different MAC based architectures, one with a RCA in the final stage and the other with a CLA in the final stage. The MSE improvement is calculated as the difference between MSE of (VOS+truncation) with and without compensation. We use four sample images (Baboon, Lena, Flight, and Pepper) and

list the average MSE improvement in Table-I. We see that the MAC with RCA has slightly higher MSE improvement. While the MSE performance of the two MAC based systems is slightly different, both benefit from use of this technique.

We compare the performance of the proposed voltage scaling with dynamic range reduction technique with the ANT technique for FIR filtering [4]. The reduced computation block in the ANT system is a filter that uses 4 MSBs for both filter coefficients and input values. It consumes approximately 23% extra energy at nominal voltage but at 0.8V, the ANT system achieves 20% energy reduction for MSE=60 noise power. In comparison, the proposed technique achieves 85% energy reduction for the same level of noise power.

TABLE-I:
MSE IMPROVEMENT FOR DIFFERENT GAUSSIAN FILTERS

	$\sigma = 0.75$	$\sigma = 1$	$\sigma = 1.25$
MAC with RCA	54%	66%	68%
MAC with CLA	51%	62%	63%

C. Combining Computation Reduction and Dynamic Range Reduction

Computation reduction and dynamic range reduction techniques both try to keep significant computations while removing less significant portions of the computation. The combination is highly dependent on quality requirement and characteristics of the application. We illustrate this method using DCT as a case study.

Here the combination is based on DCT coefficient deactivation and low order bit truncation. The DCT architecture under consideration is given in Figure 8. In DCT coefficient deactivation, DCT coefficients are deactivated starting from the highest frequency component of 1D DCT (W_7). Thus it is not possible to deactivate W_6 without deactivating W_7. In low order bit truncation, inputs are truncated for the entire computation unit with a granularity of 2-bit. These two techniques are combined in such a way that the performance degradation is minimized. Figure 17 illustrates the proposed methods for 14-bit fixed point DCT implementation. The solid red line in Figure 17 illustrates the scenario in which W_7 is deactivated and 4 low order bits are truncated in the rest of the coefficients. The above procedure can be implemented by controlling the AND gates at the inputs of each DCT coefficient computation unit as illustrated in Figure 8.

Next, we describe a scheme to combine coefficient deactivation and low order bit truncation. First we note that there is a crossover point in performance where it becomes better to deactivate a coefficient instead of applying aggressive bit truncation to that coefficient. Figure 18 illustrates the PSNR performance of Baboon image as a function of low order bit truncation for W_5 and W_7 coefficients. We see that deactivation of W_5 and W_7 coefficients become more attractive after truncating 7 bits (out of 14). To improve confidence of the crossover point, we investigate the performance of 6 sample images (Baboon, Lena, Flight, Pepper, House and Bridge) and find that it is better to deactivate the DCT coefficient rather than truncating 6 bits. Thus, in our procedure, we limit the low order bit truncation to 4 levels with granularity of 2 bits, namely 0 bit (no truncation), 2 bit, 4 bit, and 6 bit truncation.

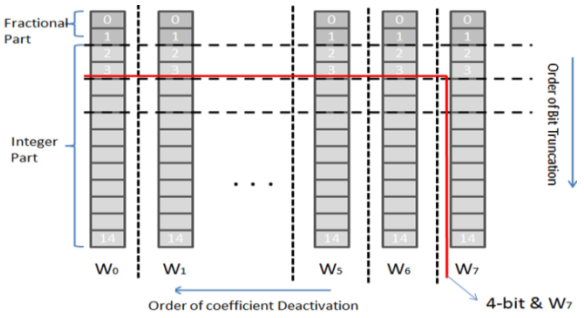


Figure 17: DCT coefficient deactivation and low order bit truncation for the 1D-DCT coefficients

Next, we determine the order in which coefficient deactivation and low order bit truncation is applied using a binary decision tree as illustrated in Figure 19. We start from full precision, and at Level 1 choose between two competing schemes 2-bit low order truncation and W_7 deactivation based on PSNR. If 2-bit truncation provides better performance, then we pick that branch. In Level 2, we choose between 4-bit truncation (2+2) and W_7 deactivation with 2-bit low order bit truncation. If in Level 1, W_7 was deactivated, then in Level 2, we choose between W_7 deactivation and 2 bit truncation of all other coefficients or W_7 and W_6 deactivation.

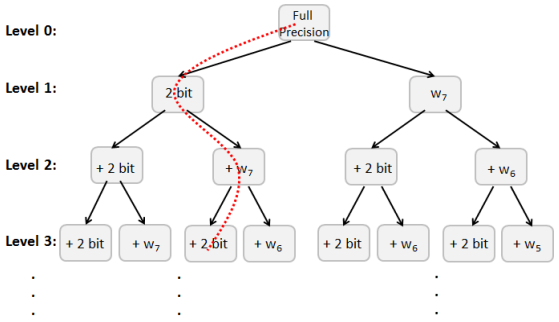
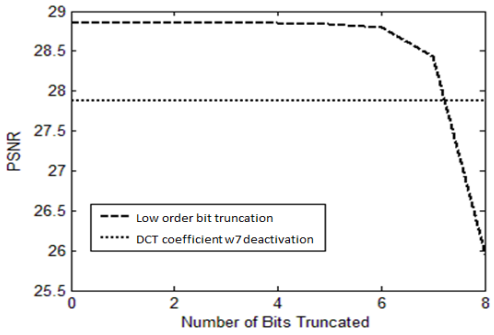
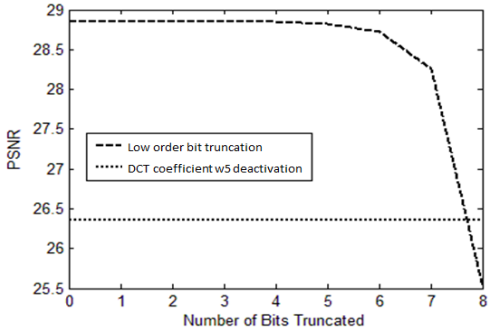


Figure 19: Binary decision tree to choose combination of coefficient deactivation and truncation level

Table II lists the reduction order of 6 images using the binary decision tree method. We read the table from left to right in increasing level number so Level 4 for Lena image corresponds to deactivation of coefficients W_6, W_7 and 2+2=4-bit truncation of all coefficients. Using majority voter scheme for each level (each column of Table II), we form a general order which is given in the last row of Table II. In this order, 2-bit low order truncation is followed by W_7 deactivation. Then, (2+2=) 4-bit low order bit truncation is applied to all the coefficients. Note that, since we consider the same computation units for the W_0 and W_4 pair to minimize the circuitry, we do not deactivate one of the members of this pairs unless both of them are eligible to be deactivated. Thus



(a)



(b)

Figure 18: Comparison between bit truncation level and DCT coefficient deactivation for a) W_7, b) W_5 for Baboon image when Q=55

TABLE II:
REDUCTION ORDER OF 6 SAMPLE IMAGES

Level	1	2	3	4	5	6	7	8
Lena	2bit	+W_7	+W_6	+2 bit	+W_5	+2bit	+W_4	+W_3
Pepper	2bit	+W_7	+2bit	+W_6	+W_5	+2bit	+W_4	+W_3
Bridge	2bit	+W_7	+2bit	+W_6	+W_5	+2bit	+W_4	+W_3
Baboon	2bit	+2bit	+W_6	+W_6	+2bit	+W_5	+W_4	+W_3
Flight	2bit	+W_7	+2bit	+W_6	+W_5	+2bit	+W_4	+W_3
House	2bit	+W_7	+2bit	+W_6	+W_5	+2bit	+W_4	+W_3
Majority	**2bit**	**+W_7**	**+2bit**	**+W_6**	**+W_5**	**+2bit**	**+W_4**	**+W_3**

Majority Voter	Reduction Technique
Level-1 (L1)	2-bit
Level-2 (L2)	2-bit & W_7
Level-3 (L3)	4-bit & W_7
Level-4 (L4)	4-bit & W_6 and W_7
Level-5 (L5)	4-bit & W_5, W_6 and W_7
Level-6 (L6)	6-bit & W_5, W_6 and W_7
Level-7 (L7)	6-bit & W_5, W_6 and W_7
Level-8 (L8)	6-bit & W_3, W_5, W_6 and W_7

in Level 7, we do not deactivate W_4. The reduction order for the first eight levels of majority voter result is also illustrated in Table II to the right.

Using Table II, we determine suitable configurations for three PSNR degradation schemes: i) Scheme-I (ΔPSNR<0.5dB), ii) Scheme-II (ΔPSNR<1dB), and iii) Scheme-III (ΔPSNR<1.5dB). Here, ΔPSNR is defined as the reduction in the PSNR value of the modified scheme compared to the baseline scheme. We use 6 sample images (Lena, Pepper, Bridge, Baboon, Flight and House) in our evaluation. For a given quality metric (Q) which is used in JPEG [50], we find all configurations that satisfy the PSNR constraint and choose the one that provides highest saving in computation. We use the majority voter order generated in Table II to determine the priority of a configuration. Table III lists the combination orders for the three schemes.

TABLE-III:
FINAL ORDER FOR SCHEMES I,II III

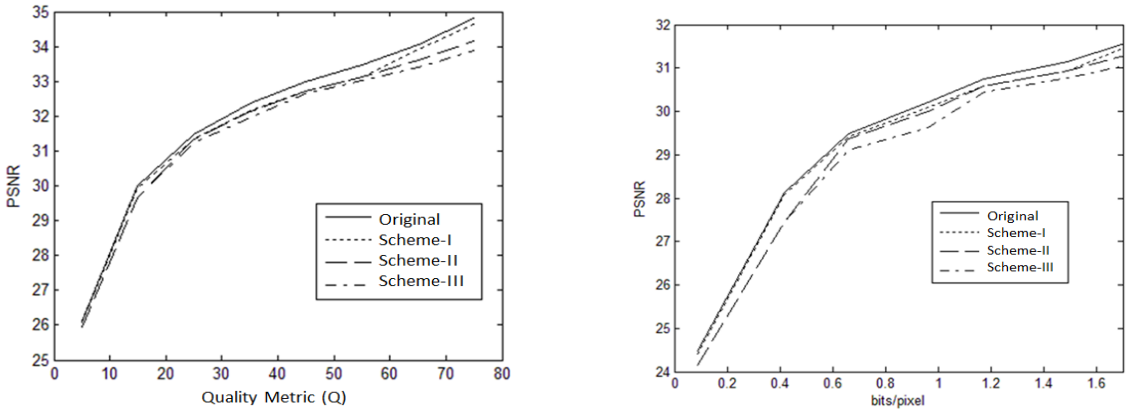
Q (Quality Metric)	75	65	55	45	35	25	15	5
Scheme-I (ΔPSNR<0.5dB)	L2	L2	L3	L3	L3	L3	L4	L6
Scheme-II (ΔPSNR<1dB)	L3	L3	L3	L3	L4	L4	L5	L8
Scheme-III (ΔPSNR<1.5dB)	L4	L4	L4	L4	L5	L5	L6	L8

We test the effectiveness of the combination schemes given in Table III using five test images (Lake, Tank, Elaine, Feather and Boat). Figure 20 illustrates the results for Elaine and Lake images. For instance, for Lake image at Q=50, Scheme II corresponding to $\Delta PSNR \leq 1dB$, results in PSNR of 32.7 dB, which is only 0.6dB lower than the original PSNR. Thus the proposed method guarantees that the PSNR constraints are satisfied for Q values from 75 down to 5.

Next, we calculate the power consumption of the original and proposed schemes for different configurations. All multiplications are implemented using carry save adder structures. Table IV lists active power, latency and area estimations for the configuration order given in Table II using 45nm PTM models. The proposed schemes have marginal increase in circuitry area (2.9%) and leakage (3.6%) compared to the original implementation due to extra units that are used to gate inputs and compensate truncation error. Overall, the proposed scheme provides flexible performance with reduced power consumption for different quality requirements. For instance, using configuration order of L8, we save 61% power consumption and have 14% extra timing slack compared to original full precision DCT engine. The timing slack can be absorbed by operating at 0.9V (instead of 1V) resulting in 68% saving in power consumption.

Finally, we present the power savings of Schemes I, II and III for Q values from 75 to 5 in Figure 21. We see that Scheme III always achieves the highest power saving due to higher allowable degradation. As we move from high quality (Q large) regions to low quality regions (Q small), we see an increase in the power savings. This is because the combinations used in the low quality regions are quite aggressive in terms of power savings. On average, we achieve 33% power saving for Scheme I, 39% power saving for Scheme II and 46% power saving for Scheme III.

We compare the proposed scheme with the ANT technique



(a) (b)

Figure 20: Performance of the resulting decision order generated using 6 training samples on test images a) Lake, and b) Elaine.

TABLE IV:
DELAY, POWER AND AREA FOR ORIGINAL AND REDUCED COMPUTATION 1D-DCT

	Original		Proposed Scheme: Reduced Computation + Truncation						
		Baseline	L1	L2	L3	L4	L5	L6	L8
Power Active (mW)	5.39	5.51	4.76	4.20	3.78	3.32	2.68	2.13	1.51
Latency(ns)	2.92	3.11	2.95	2.95	2.78	2.78	2.51	2.51	2.51
Area (um^2)	4435	4539							

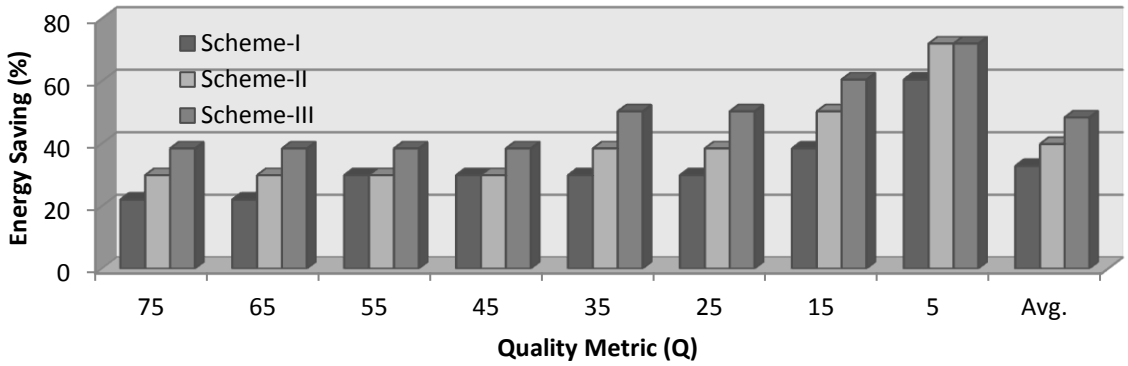


Figure 21: Energy saving at different quality levels for three candidate schemes

[5], significance driven technique in [44] and adaptive truncation technique in [23]. ANT based scheme using 4-bit MSB replica of the DCT in the reduced computation block has a 16% overhead compared to original DCT. We found that it achieves 23% energy saving with approximately 5dB degradation in PSNR. Significance driven technique achieves 47% energy reduction with more than 4dB degradation in PSNR [44]. The truncation based technique in [23] achieves up to 40% energy saving while inducing approximately 0.2 reduction in MSSM which corresponds to approximately 15dB loss in PSNR. In contrast, the proposed combination scheme can achieve average energy savings of 46% with <1.5dB PSNR degradation. Moreover, the proposed scheme provides a mechanism for higher energy saving for low Q settings while keeping the degradation low for high Q settings.

IV. Conclusion

In this paper, we presented several general as well as algorithm-specific techniques that trade-off energy with system performance for multimedia signal processing algorithms. We provided an overview of energy-savings techniques such as voltage scaling, reducing number of computations and reducing dynamic range. All these techniques introduce errors which can be compensated by algorithm-level optimizations.

Next, we described several hybrid techniques that further reduce energy consumption while causing little reduction in quality. We investigated the combination of voltage scaling and dynamic range reduction and applied it to a low pass FIR filter. The proposed scheme achieved 85% energy saving for fairly low noise level. We also studied the combination of

computation reduction and dynamic range reduction for DCT used in JPEG. Simulation results showed, on average, 33% to 46% reduction in energy consumption for a small 0.5dB to 1.5dB degradation in the system performance. Thus algorithm-level optimizations can help reduce the energy consumption of many multimedia signal processing algorithms with only a mild degradation in quality.

APPENDIX

Here, we present the expected variance of error due to truncation for unsigned multiplication. Here, two inputs $x[M:0]$ and $y[M:0]$ are independent and L bit truncation is applied before multiplication to obtain $x[M:L]$ and $y[M:L]$.

$$E[Q_{mul}(L-1:0)] = E[x(M:0)y(M:0) - x(M:L)y(M:L)] \\ = 2\mu_q(M:0)\mu_q(L-1:0) - \mu_q(L-1:0)^2$$

$$var[Q_{mul}(L-1:0)] = var[x(M:0)y(M:0) - x(M:L)y(M:L)] \\ = var[x(M:L)y(L-1:0) + x(L-1:0)y(M:L) \\ + x(L-1:0)y(L-1:0)] \\ = var[x(M:L)y(L-1:0)] + var[x(L-1:0)y(M:L)] \\ +2\{cov[x(M:L)y(L-1:0), x(L-1:0)y(M:L)] \\ +cov[x(M:L)y(L-1:0), x(L-1:0)y(L-1:0)] \\ +cov[x(L-1:0)y(M:L), x(L-1:0)y(L-1:0)]\} \\ +var[x(L-1:0)y(L-1:0)]$$

where $cov(.,.)$ represents the covariance operation. We use the variance and covariance property for product of random variables to simplify the above expression.

$$var[Q_{mul}(L-1:0)] = 2[\mu_q(L-1:0)^2\sigma_q^2(M:L) \\ + \mu_q(M:L)^2\sigma_q^2(L-1:0) + \sigma_q^2(L-1:0)\,\sigma_q^2(M:L)] \\ +2\mu_q(L-1:0)^2\sigma_q^2(L-1:0) + \sigma_q^2(L-1:0)\sigma_q^2(L-1:0) \\ +2\mu_q(L-1:0)\mu_q(M:L)\sigma_q^2(L-1:0).$$

REFERENCES

[1] S. Ghosh and K. Roy, "Parameter variation tolerance and error resiliency: new design paradigm for the nanoscale era," Proceedings of the IEEE, pp. 1718-1751, Oct. 2010

[2] S. H. Nawab, A. V. Oppenheim, A. Chandrakasan, J. M Winograd, and J. T. Ludwig, "Approximate signal processing," Journal of VLSI Signal Processing, vol. 15, pp. 177- 200, 1997.

[3] A. Chandrakasan and R. Brodersen, "Minimizing power consumption in digital CMOS circuits," Proceedings of the IEEE, pp. 498-523, April, 1995.

[4] R. Hegde and N. R. Shanbhag, "Soft digital signal processing," IEEE Trans. on VLSI Systems, vol. 9, no. 12, pp. 813–823, Dec. 2001.

[5] B. Shim, S. R. Sridhara and Naresh R. Shanbhag, "Reliable low-power digital signal processing via reduced precision redundancy," IEEE Trans. on VLSI Systems, vol. 12, pp. 497-510, May 2004.

[6] G. Varatkar and N. R. Shanbhag, "Error-resilient motion estimation architecture," IEEE Trans. on VLSI Systems, vol. 16, no. 10, pp. 1399–1412, Oct. 2008.

[7] N. R. Shanbhag, R. A. Abdallah, R. Kumar and D. L. Jones, "Stochastic computation," Design Automation Conference, pp. 859-864, June 2010.

[8] Y. Emre and C. Chakrabarti, "Data-path and memory error compensation technique for low power JPEG implementation," Int. Conf. on Acoustics Speech and Signal Processing, pp. 1589-1592, May 2011.

[9] Y. Emre and C. Chakrabarti, "Memory error compensation techniques for JPEG2000," IEEE Workshop on Signal Processing Systems, pp. 36-41, Oct. 2010.

[10] I. J. Chang, D. Mohapatra and K. Roy, "A voltage-scalable & process variation resilient hybrid SRAM architecture for MPEG-4 video processors," Design and Automation Conference, pp. 670-675, 2009.

[11] M. Cho, J. Schlessman, W. Wolf and S. Mukhopadhyay, "Accuracy-aware SRAM: a reconfigurable low power SRAM architecture for mobile multimedia applications," Asia and South Pacific Design Automation Conference, pp. 823-828, 2009.

[12] Z. Chishti and et al., "Improving cache lifetime reliability at ultra-low voltages," Int. Symp. on Microarchitecture, pp. 89-99, Dec. 2009.

[13] M. A. Makhzan, A. Khajeh, A. Eltawil and F. J. Kurdahi, "A low power JPEG2000 encoder with iterative and fault tolerant error concealment," IEEE Trans. on VLSI Systems, vol. 17, no 6, pp. 827-837, June 2009.

[14] J. Sartori and R. Kumar, "Architecting processors to allow voltage/reliability tradeoffs," Int. Conf. on Compiler Architectures and Synthesis for Embedded Systems, pp. 115-124, Oct. 2011.

[15] K. J. Lin, S. Natarajan and J. W. S. Liu, "Imprecise results: utilizing partial computations in real-time systems," in Proc. 8th Real-Time System Symposium, pp. 210-217, Dec 1987.

[16] A. Sinha, A. Wang, and A. Chandrakasan, "Energy scalable system design," IEEE Trans. on VLSI Systems, vol 10, pp.135-145, April 2002.

[17] C. Chen and et al., "Analysis and architecture design of variable block-size motion estimation for H.264/AVC," IEEE Trans. on Circuit and System-I, vol. 53, no. 2, pp. 578-593, Feb. 2006.

[18] Y. Emre and C. Chakrabarti, "Low energy motion estimation via selective approximation," IEEE Int. Conf. on Application-Specific Systems, Architectures and Processors, pp. 176-183, Sept. 2011.

[19] C. Lian and et al., "Power-aware multimedia: concepts and challenges," IEEE Circuits and Systems magazine, vol. 7, no. 2, pp 26-34, 2007.

[20] Y. Andreopoulos and M. van der Schaar, "Incremental refinement of computation for the discrete wavelet transform," IEEE Trans. on Signal Processing, vol. 56, pp. 140-157, Jan. 2008.

[21] J. Y. F. Tong, D. Nagle and R. A. Rutenbar, "Reducing power by optimizing the necessary precision/range of floating point arithmetic," IEEE Trans. on VLSI Systems, vol. 8, pp. 273-286 June 2000.

[22] J. Park, J. H. Choi and K. Roy, "Dynamic bit-width adaptation in DCT: an approach to trade-off image quality and computation energy," IEEE Trans. on VLSI Systems, vol. 18, pp. 787-793, May 2010.

[23] S. H. Kim, S. Mukohopadhyay and M. Wolf, "System level energy optimization for error tolerant image compression," IEEE Embedded System Letters, vol. 2, pp. 81-84, Sept. 2010.

[24] Z. He, C. Tsui, K. Chan, and M.L. Liou, "Low-power VLSI design for motion estimation using adaptive pixel truncation," IEEE Trans. on Circuits and Systems for Video Technology, vol. 10, no 5, pp. 669-678, August 2000.

[25] D. Ernst and et al., "Razor: a low-power pipeline based on circuit-level timing speculation," Int. Symp. on Microarchitecture, pp. 7-18, Dec 2003.

[26] K. A. Bowman, J. W. Tschanz, N.S. Kim, J. C. Lee, C. B. Wilkerson, S. L. Lu, T. Karnik, and V. K. De, "Energy-efficient and metastability-immune resilient circuits for dynamic variation tolerance," IEEE Journal of Solid State Circuits, vol 44, no 1, pp. 49-63, Jan 2009.

[27] Predictive Technology Modeling: ptm.asu.edu

[28] S. Mukhopadhyay, H. Mahmoodi and K. Roy, "Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS," IEEE Trans. on CAD of Integrated Circuits and Systems, vol. 24, no 12, pp. 1859-1880, Dec. 2005.

[29] K. Agarwal, and S. Nassif, "Statistical analysis of SRAM cell stability," Design Automation Conference, pp. 57-62, Sept 2006.

[30] K. Zhang, "Embedded memories for nanoscale VLSIs," Springer 2009

[31] G. K. Chen and et al., "Yield-driven near-threshold SRAM design," in Int. Conf. Computer Aided Design, pp. 660–666, 2007.

[32] Y. Emre and C. Chakrabarti "Techniques for compensating memory errors in JPEG2000," will appear in IEEE Trans. on VLSI Systems.

[33] T.N. Rao, and E.Fujiwara "Error control coding for computer systems," Prentice Hall, 1989.

[34] J. Kim and et al., "Multi-bit tolerant caches using two dimensional error coding," Int. Symp. on Microarchitecture, pp. 197-209, 2007.

[35] F. J. Kurdahi, A. Eltawil, K. Yi, S. Cheng and A. Khajeh, "Low-power multimedia system design by aggressive voltage scaling," IEEE Trans. VLSI Syst., vol 18, no 5, pp. 852-856, May 2010.

[36] M. Goel and N.R. Shanbhag, "Dynamic algorithm transforms for low-power adaptive equalizers," IEEE Transactions on Signal Processing, vol. 47, no. 10, pp. 2821–2832, Oct. 1999.

[37] Y. Andreopoulos, and I. Patras, "Incremental Refinement of Image Salient-Point Detection," IEEE Trans. on Image Processing, vol 17, pp 1685-1699, Sept 2008.

[38] A. K. Verma, P. Brisk and P. Ienne, "Variable latency speculative addition: a new paradigm for arithmetic circuit design," Design, Automation and Test in Europe (DATE), pp. 1250-1255, 2008.

[39] N. Zhu, W. Goh, W. Zhang, K. Yeo and Z. Kong, "Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing", IEEE Trans. on VLSI Systems, vol. 18, no.8, pp. 1225–1229 August 2010.

[40] M.D. Ercegovac and T. Lang, "Digital arithmetic", Morgan Kauffmann Publishers, 2004.

[41] M. R. Chowdhury and K. Mohanram, "Approximate logic circuits for low overhead, non-intrusive concurrent error detection," Design, Automation and Test Conference in Europe (DATE), pp. 903-908, 2009.

[42] P. Kulkarni, P. Gupta, and M. Ercegovac, "Trading accuracy for power in a multiplier architecture," Journal of Low Power Electronics, pp. 490-501, Dec 2011.

[43] D. Shin and S. K. Gupta, "Approximate logic synthesis for error tolerant applications," in Proc. 13th IEEE Design, Automation and Test in Europe, pp. 957-960, March 2010.

[44] G. Karakonstantis, N. Banerjee and K. Roy, " Process-variation resilient and voltage scalable DCT architecture for robust low-power computing," IEEE Trans. on VLSI Systems, vol. 18, no. 10, pp. 1461–1470, Oct. 2010.

[45] N. Banerjee and et al., "Design methodology for low power dissipation and parametric robustness through output quality modulation: application to color interpolation filtering," IEEE Trans. CAD for Integrated Circuits and Systems, vol. 28, no. 8, pp. 1127–1137, Aug. 2009.

[46] D. Mohapatra, G. Karakonstantis, and K. Roy, "Significance driven computation: a voltage-scalable, variation-aware, quality-tuning motion estimator," in Int. Symp. Low Power Electronics and Design, pp. 195–200, 2009.

[47] V. Chippa, D. Mohapatra, A. Raghunathan, K. Roy, and S. T. Chakradhar, "Scalable effort hardware design: exploiting algorithmic resiliency for energy efficiency," Design Automation Conference, pp. 555-560, June 2010.

[48] J. George, B. Marr, B. E. S. Akgul, and K. V. Palem, "Probabilistic arithmetic and energy efficient embedded signal processing," in International Conference on Compilers Architectures and Syhthesis for Embedded Systems (CASES), pp. 158-168, 2006.

[49] Y. Liu, and T. Zhang, **"**On the selection of arithmetic unit structure in voltage overscaled soft digital signal processing", International Symposium on Low Power Electronics and Design (ISLPED), pp. 250-255, Aug, 2007.

[50] T. Acharya and P.-S. Tsai, "JPEG2000 standard for image compression: concepts, algorithms and VLSI architectures," Wiley Inter-Science, 2004.