J Sign Process Syst
DOI 10.1007/s11265-012-0667-5

Quality-Aware Techniques for Reducing Power of JPEG Codecs

Yunus Emre · Chaitali Chakrabarti

Received: 4 November 2011 / Revised: 30 January 2012 / Accepted: 8 February 2012

Abstract This paper presents use of bit truncation and voltage overscaling to reduce the power consumption of JPEG codecs. Both techniques introduce errors which have to be compensated to minimize quality degradation. To handle the errors due to bit truncation, we propose a compensation scheme based on unbiased estimation of the truncation noise. For 4-bit truncation, such a scheme achieves 23% power savings for DCT with only 0.6 dB drop in PSNR. To compensate for errors due to aggressive voltage scaling, we introduce an algorithm-specific technique which is based on exploiting the characteristics of the quantized coefficients after zig-zag scan. This technique is very effective in improving the PSNR performance with a small circuit overhead. A combination of the two techniques help achieve even higher power savings with only a modest increase in PSNR. For instance, a combination of 4-bit truncation and operating voltage of 0.78 V results in 44% power reduction for DCT with a 1.8 dB drop in PSNR performance of the JPEG codec.

Keywords JPEG · Truncation · Voltage scaling · Error compensation

This work was funded in part by NSF CSR0910699.

Y. Emre (✉) · C. Chakrabarti
School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, USA
e-mail: yemre@asu.edu

C. Chakrabarti
e-mail: chaitali@asu.edu

1 Introduction

JPEG is one of the most widely used image compression standards today. It has slightly lower compression performance compared to JPEG2000, but because of its simple structure and ease of implementation, it is still very popular. JPEG is part of many embedded devices for multimedia where power consumption is a very important metric. An effective way of reducing the power consumption of these devices is lowering the supply voltage. However, this could result in critical path violations leading to failures. Operating on a narrower datapath by truncating the lower order bits also helps reduce the power consumption but introduces truncation errors. Thus these power saving methods cannot be directly used for high quality imaging applications. This paper describes methods to compensate for the errors caused by truncation and aggressive voltage scaling and provides a mechanism for lowering power with only a mild degradation in quality.

Several JPEG architectures have been proposed that trade-off power consumption and quality. They primarily focus on discrete cosine transform (DCT) which is one of the high power consuming units [1–4]. The DCT architecture in [1] exploits correlation between DCT coefficients in conjunction with standard techniques such as voltage scaling, data parallelism and pipelining. Data bit-width adaptation is used in [2] to reduce the processing load of high frequency cefficient computations. A similar scheme is also investigated in [3] where truncation of up to 4 low order bits achieves 40% reduction in energy consumption of the memory and data-path. Process variations effects are considered in [4] which generates the more important DCT coefficients first and uses longer delay paths for the

Published online: 03 March 2012

less important coefficients. Algorithmic noise tolerance and N-modular redundancy techniques are investigated for DCT based image coding system in [5]. In [6], an analysis of the relation between input image characteristics and operating voltage for low energy systems is presented.

Memory, power and image quality trade-offs have been studied in [7] where memory banks that store most significant bits (MSB) are operated at a different voltage level than the ones that store less significant bits (LSB), thereby achieving power reduction with some degradation in image quality. In [8], for higher reliability in low voltage operation, MSBs are stored in a memory bank with 8T SRAM cells and the LSBs are stored in banks with 6T SRAM cells. More recently, algorithm-specific techniques to mitigate the effects of SRAM memory failures caused by low voltage operation in JPEG2000 implementations have been proposed in [13].

In this work, we investigate use of bit truncation and voltage overscaling to reduce the power consumption of JPEG codecs with minimal effect on the image quality. Since both these methods introduce errors, we propose compensation techniques with low overhead to mitigate the effect of these errors. To compensate for errors due to truncation, we use an unbiased estimator based technique. For 4-bit truncation, this results in 23% power savings for DCT with only 0.6dB drop in peak signal to power ratio (PSNR). To compensate for errors due to aggressive voltage scaling, we introduce an algorithm-specific technique first proposed in [9]. The technique exploits the fact that in 8×8 DCT, two adjacent AC coefficients after zig-zag scan have similar values and two coefficients corresponding to higher frequencies generally have smaller values. These features are used to detect the datapath errors and then compensate. Operating the datapath at 0.83 V (instead of the nominal 1 V), results in BER= 10^{-4} due to voltage overscaling. For this error rate, the proposed technique achieves 3.4 dB PSNR improvement compared to no correction case and approximately 1.2 dB degradation compared to error-free performance for a 20% reduction in power consumption. A combination of bit truncation and voltage overscaling techniques helps achieve even higher power savings. For instance, for 0.78 V operating voltage and 4-bit truncation, the power reduction is as high as 44% with a 1.8 dB drop in PSNR. Thus the proposed techniques enable JPEG codecs to have much lower power consumption with only a mild degradation in image quality.

The paper is organized as follows. We present a brief description of JPEG in Section 2, followed by analysis of reduced precision and a technique for compensating the associated errors in Section 3. Analysis of failures due to voltage overscaling and the corresponding compensation technique is presented in Section 4. Simulation results illustrating the performance of the techniques and synthesis results of overhead circuitry are described in Section 5. The paper is concluded in Section 6.

2 Background

The general block diagram of a JPEG encoder/decoder is shown in Fig. 1. The original image in pixel domain is divided into 8×8 blocks which are transformed into frequency domain using 2 dimensional (2-D) DCT. This is followed by quantization, where the coefficients are scaled by factors that depend on the desired image quality and/or compression rate. Next, zig-zag scanning is used to order the 8×8 quantized coefficients into a one dimensional vector (1×64 format) where low frequency coefficients are placed before the high frequency coefficients. The entropy coder generates the compressed image using Huffman coding.

Discrete Cosine Transform 2-D DCT is typically implemented using 1-D DCTs along rows (columns) followed by 1-D DCT along columns (rows) as illustrated in Fig. 2. The transpose unit helps in getting the data in the right order for the second 1D DCT unit.

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

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

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 stored in the transpose unit 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. We use the following method for implementing the odd and even coefficients.

$$\begin{bmatrix} w_0 \\ w_2 \\ w_4 \\ w_6 \end{bmatrix} = \begin{bmatrix} d & d & d & d \\ b & f & -f & -b \\ d & -d & -d & d \\ f & -b & b & -f \end{bmatrix} \begin{bmatrix} x_0 + x_7 \\ x_1 + x_6 \\ x_2 + x_5 \\ x_3 + x_4 \end{bmatrix} \tag{2}$$

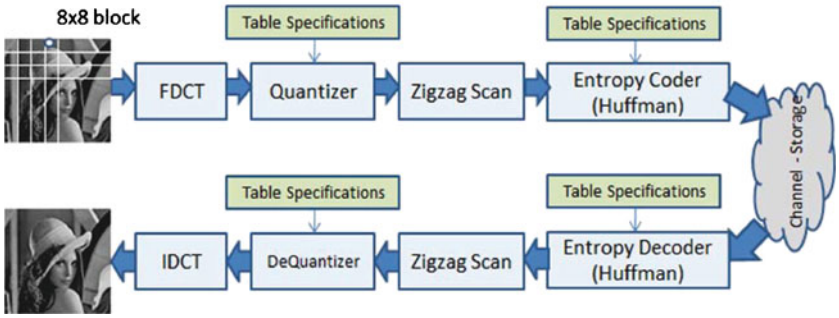


Figure 1 Block diagram of JPEG.

$$\begin{bmatrix} w_1 \\ w_3 \\ w_5 \\ w_7 \end{bmatrix} = \begin{bmatrix} a & c & e & g \\ c & -g & -a & -e \\ e & -a & g & c \\ g & -e & c & -a \end{bmatrix} \begin{bmatrix} x_0 - x_7 \\ x_1 - x_6 \\ x_2 - x_5 \\ x_3 - x_4 \end{bmatrix} \tag{3}$$

where $a = \frac{1}{2}\cos(\frac{\pi}{16})$, $b = \frac{1}{2}\cos(\frac{2\pi}{16})$, $c = \frac{1}{2}\cos(\frac{3\pi}{16})$, $d = \frac{1}{2}\cos(\frac{4\pi}{16})$, $e = \frac{1}{2}\cos(\frac{5\pi}{16})$, $f = \frac{1}{2}\cos(\frac{6\pi}{16})$, $g = \frac{1}{2}\cos(\frac{7\pi}{16})$. The DCT engine is implemented by 12 bit integer operations in [2, 10]. However, in our analysis, we introduce 2 extra bits to represent the fractional part of the computation in baseline mode. This results in approximately 0.1dB improvement over the 12-bit implementation.

The architecture of 4 DCT coefficients (w_0, w_1, w_2 and w_4) are illustrated in Fig. 3. For w_0 and w_4, common sub-expression elimination (CSE) is used to obtain results with small number of computation units (see Fig. 3(b)). Implementation of w_2 is illustrated in Fig. 3(c); a variant of which is used for w_6. Figure 3(d) shows the computation structure used to find w_1. The odd coefficients, w_3, w_5, w_7, are computed using units that are similar to the unit for w_1. All multiplications are implemented with shifters and adders. The critical path is that of a 8-input carry save adder (CSA) tree.

Quantizer The rate and quality of the image is determined at the quantizer. In order to achieve different quality and compression rates, the quantization matrix is multiplied with a quality factor that is determined with the help of quality metric (Q) which ranges from 1 to 100 [11]. A lower Q result in lower image quality and higher compression rate. Figure 4(a) illustrates JPEG luminance quantization table for Q=50. Note that high frequency components which are at the bottom right corner are quantized aggressively while low frequency components which are at the top left corner are mildly quantized. Figure 4(b) also shows the zig-zag scanning order. The very first element is the DC coefficient which is encoded in differential order by subtracting the DC coefficient of the previous block and encoding the difference using a Huffman table in baseline JPEG; the rest of the coefficients are AC coefficients, which are encoded using another Huffman table.

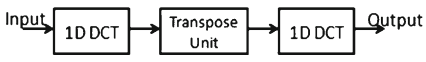


Figure 2 2D DCT architecture using 1-D DCTs.

3 Power Reduction by Truncation

Reduced precision arithmetic, which simply truncates the lower significant bits (LSB) of the inputs, is an effective method to reduce power consumption. Operating on lower number of bits results in lower critical path delay. This in turn enables operation at scaled voltage levels without critical path violation. While this method results in significant power reduction, it introduces errors and causes quality degradation.

Figure 5 illustrates the timing slack and savings in power consumption of a 16-bit ripple carry adder (RCA) for different bit widths. The adder was implemented using 45 nm PTM models (ptm.asu.edu) and Monte Carlo simulations were run to generate these results. Since RCA has a regular structure, the power reduction and timing slack are both proportional to the bit-width of the adder. For instance, at nominal voltage, we observe 28% reduction in power consumption when we use 12-bit precision instead of 16-bits. The higher the truncation order, higher is the power savings, as expected. However such a scheme introduces truncation errors that have to be compensated to avoid noticeable quality degradation.

3.1 Truncation Induced Error

First, we investigate the effect of bit truncation on simple adder operation. Then in Section 3.2, we describe a method to compensate for these errors. Let us consider a system whose inputs are originally represented with $M+1$ bits, $x(M:0)$. When L bit truncation is

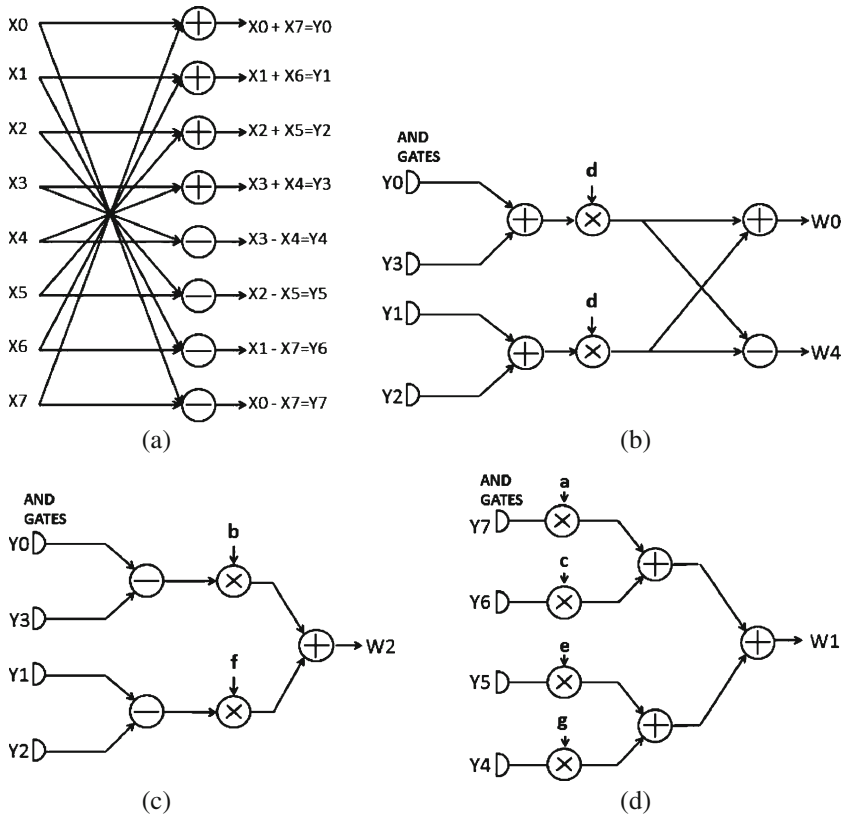


Figure 3 Architecture of 1-D DCT coefficients. (**a**) First stage butterfly (**b**) w_0 and w_4 computation units, (**c**) w_2 unit, (**d**) w_1 unit.

employed, where $L \leq M$, the input becomes $x(M:L)$. Assuming uniformly distributed input signals, we can express the expected truncation error for the input signal x as:

$$q_x = x(M:0) - x(M:L),$$
$$E[q_x] = E[x(L-1:0)] = \frac{2^L - 1}{2} \tag{4}$$

The truncation error (q_{add}) of an adder with inputs x and y can be expressed as:

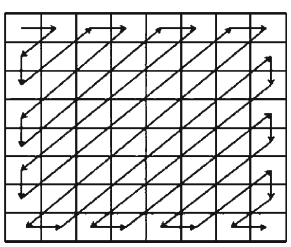
$$E[q_{\text{add}}] = E[(x(M:0) + y(M:0)) - (x(M:L) + y(M:L))]$$

If we assume that both the inputs are independent and uniformly distributed, we can express the result as:

$$E[q_{\text{add}}] = E[x(L-1:0) + y(L-1:0)] = 2 \times E[x(L-1:0)] = 2^L - 1 \tag{5}$$

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

(a)



(b)

Figure 4 (**a**) Luminance quantization matrix for Q=50; (**b**) Zigzag scan order for a 8×8 block.

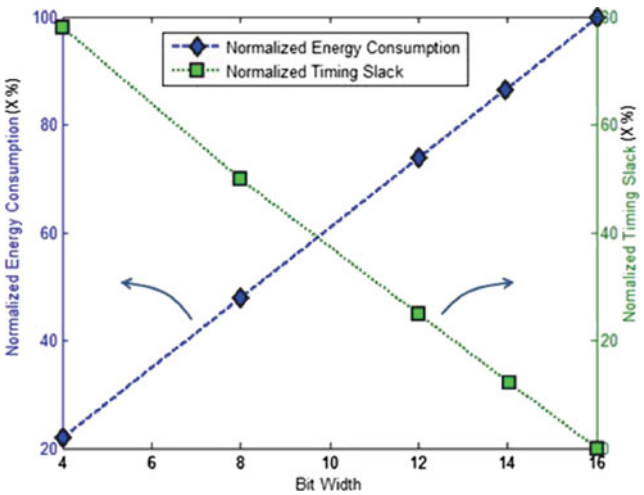


Figure 5 Energy delay distributions of RCA as a function of bit-width.

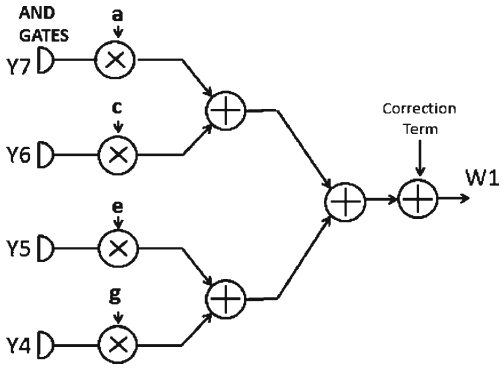


Figure 6 Processing unit for w_1 with compensation.

Using the same analysis, the expected truncation noise for a subtraction operation is given by

$$E[q_{\text{sub}}] = E[x(L-1:0) - y(L-1:0)] = 0 \tag{6}$$

3.2 Truncation Error Compensation

We use the above technique to calculate the truncation error (TN) of the DCT outputs for the architecture described in Fig. 3. The data is represented by 14 bits with 12 bits for the integer part and 2 bits for the fractional part. The expected errors due to truncation in w_0 and w_1 are derived below. Because of the 2 extra fractional bits, the expected error in Eq. 4 is normalized by $\frac{1}{4}$. To simplify our analysis, we assume that all Y values in Fig. 3, namely, Y0, Y1, Y2, Y3, are uncorrelated and so the expected value for L bit truncation is $\frac{2^L-1}{8}$. Since $w_0 = d \times (Y0 + Y1 + Y2 + Y3)$, the truncation error for w_0, is given by

$$\begin{aligned} TN_{w_0} &= E[d \times (Y0(L-1:0) + ... + Y3(L-1:0))] \\ &= \left\lfloor \frac{d(2^L-1)}{2} \right\rfloor \end{aligned} \tag{7}$$

Similarly the truncation error for w_1 is given by

$$TN_{w1} = \left\lfloor (a+c+e+g)\frac{(2^L-1)}{8} \right\rfloor \tag{8}$$

and that of w_2 is given by $TN_{w_2} = (b + f - b - f) \times E[Y] = 0$. In a similar way, TN_{w_4} and TN_{w_6} are also zero.

The expected truncation noise values are used as unbiased estimators to compensate the error. Instead of compensating for errors in 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 low. The data-paths of w_0 and w_1 units are modified by adding an adder in the last stage. Figure 6 illustrates the compensation mechanism for the w_1 computation unit. 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.

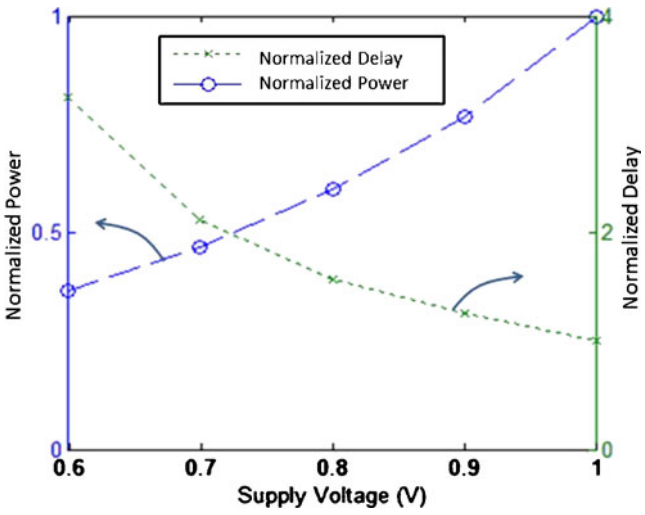


Figure 7 Energy delay profile of 14-bit RCA adder under voltage scaling.

4 Power Reduction by Voltage Scaling

Voltage scaling is one of the most effective techniques to reduce active power consumption. However, it increases the latency of the circuitry and promotes delay induced errors. Figure 7 illustrates the normalized power saving and delay increase of the 14-bit ripple carry adder (RCA) with respect to nominal voltage using 45nm PTM models (ptm.asu.edu). When the voltage is scaled to 0.8V, there is an approximately 40% reduction in power consumption of the adder and a 46% increase in the delay. Thus aggressive voltage scaling can lead to timing violations.

4.1 Voltage Scaling Induced Errors

In this section, we focus on failures in the data path which can happen because of critical path violation due

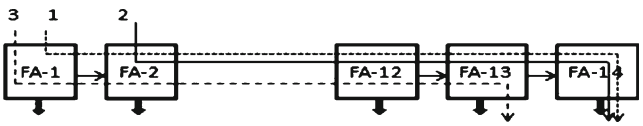


Figure 8 Block diagram of 14-bit RCA.

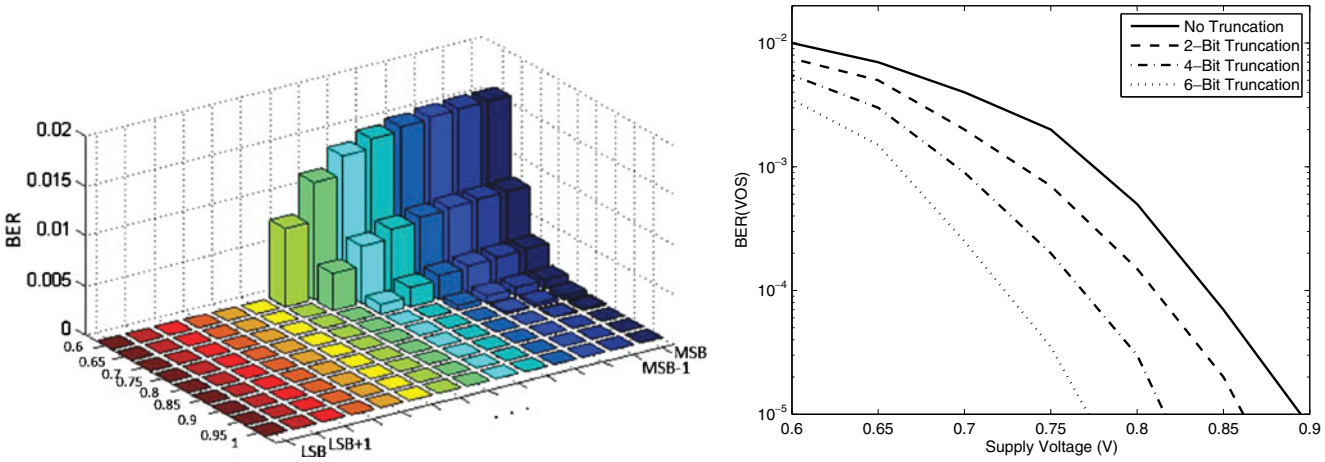


Figure 9 Probability of error distribution for 14-bit RCA for (**a**) different voltage settings, (**b**) different levels of truncation.

to aggressive voltage scaling during computation of 2D DCT followed by quantization. Assume that a single datapath violation occurs during 1D DCT along rows that result in a single miscalculated coefficient. This failure affects the values of eight 2D-DCT coefficients along a column of 8×8 DCT. Fortunately, after zig-zag scan, the miscalculated coefficients in a column are separated.

We use the method in [9] to derive the error probability distribution of a 14-bit RCA and use the results to generate the error models under voltage scaling. The 14-bit RCA is illustrated in Fig. 8, where 3 of the longer paths are highlighted.

Assume that the delay of each full adder (FA) is the sum of nominal delay, t_{FA}, systematic variation t_{SYS}, which is typically considered same for all the FAs in a 14-bit RCA, and random variation $t_{r,-}$ which can be modeled using zero mean iid Gaussian random variable with variance σ_{FA}. Then delay of each carry chain starting from the x^{th} FA and ending at the y^{th} FA can be calculated as

$$T_{\text{chain}}(x, y) = (x - y) \times (t_{FA} + t_{SYS}) + (t_{r,x} + ... + t_{r,y}) \tag{9}$$

which can be simplified using the iid Gaussian properties as:

$$T_{\text{chain}}(\Delta) = \Delta \times (t_{FA} + t_{SYS}) + \sqrt{\Delta} \times t_r \tag{10}$$

where $\Delta = x - y$. Thus $T_{\text{chain}}(\Delta)$ is a Gaussian variable with $\mu = \Delta \times (t_{FA} + t_{SYS})$ and $\sigma = \sqrt{\Delta} \times \sigma_{FA}$. Also, the delay of any chain can be represented using only 14 different distributions $T_{\text{chain}}(1)$ to $T_{\text{chain}}(14)$.

The probability of errors for each bit at the output of the 14-bit adder is derived as follows. Assume that the critical path delay is t_{crt}. We have 14 different paths that may lead to MSB error over the carry chain: LSB to MSB, $LSB+1$ to MSB, $LSB+2$ to MSB etc, where each has a different delay distribution. In order to calculate the probability of error for MSB, we use the Bayes' theorem and sum all the probabilities as:

$$p(t_{MSB} > t_{crt}) = \sum_{z=1}^{14} p(T_{\text{chain}}(z) > t_{crt} | chain = z) \times p(chain = z) \tag{11}$$

where t_{MSB} is the path delay of MSB bit and $p(chain = z) = \frac{1}{2^z}$.

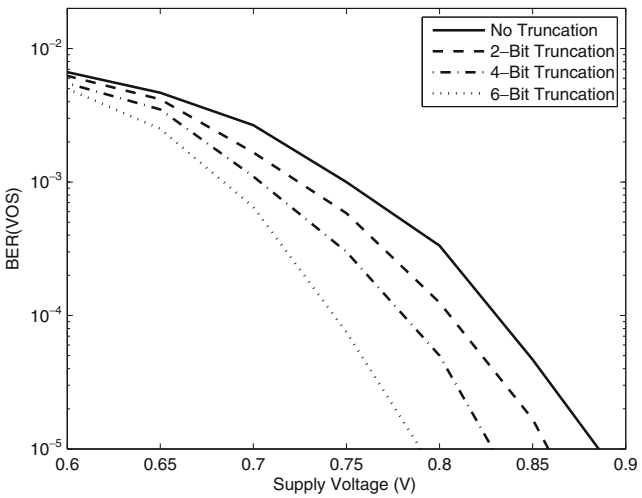


Figure 10 BER(VOS) vs supply voltage of a 8 input 14 bit carry save adder tree.

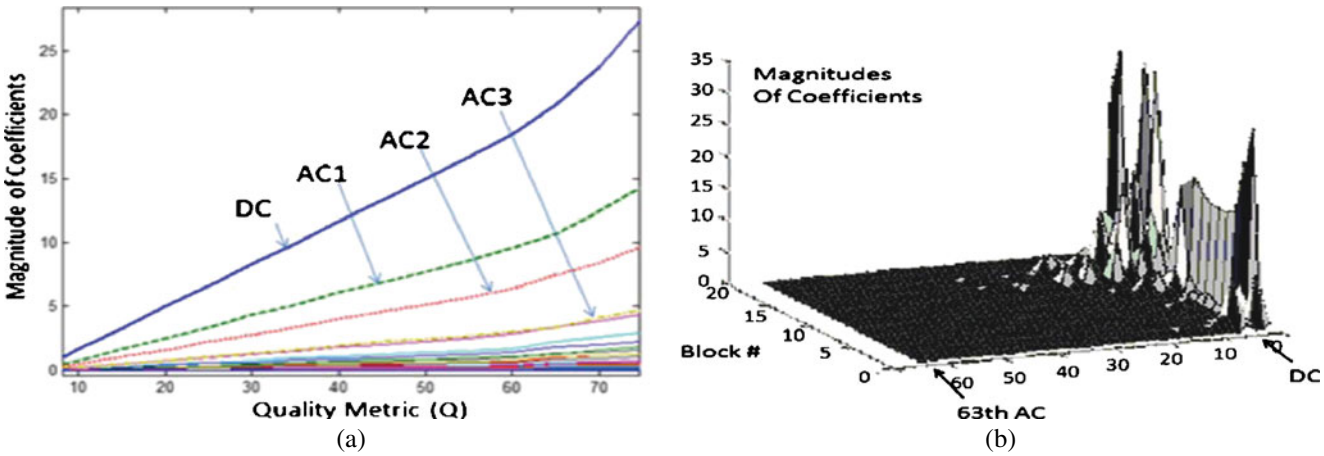


Figure 11 Magnitude of DC and AC coefficients (**a**) averaged over all blocks; (**b**) first 20 blocks of Bridge image.

Thus for each output bit we can calculate its error probability for a given t_{crt}. The distribution of errors due to voltage scaling for different supply voltages is shown in Fig. 9(a) when the allowable critical path is 1350ps. The distribution is consistent with that in [12]. The following parameters are used to obtain the distribution. At nominal voltage of 1V, $t_{FA} = 82ps$, $t_{SYS} = 5ps$ and $\sigma_{FA} = 8ps$ for fan-out of four (FO4); at 0.6V, the values increase to $t_{FA} = 240ps\, t_{SYS} = 5ps$ and $\sigma_{FA} = 15ps$.

Figure 9(b) illustrates the BER of the adder due to voltage overscaling (VOS) for different levels of truncation. Since the critical path is now lower, delay violations are also lower resulting in decrease in voltage scaling induced errors for the same supply voltage. For instance, while no-truncation achieves BER(VOS)= 10^{-4} at 0.85 V, 2-bit truncation has the same BER at 0.82 V. Note that the BER reported here is due to voltage scaling only and does not include the truncation errors that were presented in Section 3.

The same procedure can be applied to generate the BER(VOS) vs supply voltage curves for the CSA tree structures that are used to implement the DCT datapath. Figure 10 illustrates the BER(VOS) of the eight input CSA tree for different levels of truncation. A BER(VOS) of 10^{-4} can be achieved by operating at 0.83 V with no truncation and also at 0.78 V with 4-bit truncation. Later in our evaluation of the differen techniques in Section 5.3, we use these curves to get the operating voltage for different BER(VOS) and truncation levels.

4.2 Compensation for Voltage Scaling Induced Errors

In order to compensate for voltage scaling induced errors, we use algorithm-specific techniques [9]. We utilize the fact that in frequency domain, neighboring coefficients have similar values. Figure 11(a) shows the average magnitude of the DC coefficient and several AC coefficients after zig-zag scan for different values of Q for Bridge image. These figures demonstrate that (i) there is a similarity in the magnitude between coefficients of two adjacent AC coefficients after zig-zag scan, (ii) coefficients corresponding to higher frequencies generally consist of smaller values and (iii) the magnitude of coefficients increase with Q. In addition, from our simulations, we find that coefficients of the same order but in consecutive blocks also have similar magnitudes. This is illustrated in Fig. 11(b) which shows 64 coefficient values of the first 20 blocks of Bridge image when Q=50.

Recall that while the 8×8 DCT units generates 14 bit outputs, the quantization stage determines the number of bits that are finally used to represent each coefficient. For instance, when Q=50, the 5th AC (AC5) coefficient which is originally 14 bits (12 bits integer + 2 bits fractional) is quantized and rounded to $AC_q(5) = round(\frac{AC5}{10})$ which is represented with 9-bits (bold in Table 1). Table 1 specifies how many bits are sufficient to represent the coefficients after quantization step for different values of Q. In order to reduce the complexity, we partitioned the 64 coefficients into 4

Table 1 Number of bits necessary to represent each group of 2D DCT coefficients for natural images.

Quantizer	Group-1	Group-2	Group-3	Group-4
$Q \leq 5$	6	5	4	3
$5 < Q \leq 15$	7	6	5	4
$15 < Q \leq 30$	8	7	6	4
$30 < Q \leq 55$	9	8	7	6
$55 < Q \leq 70$	9	8	7	7

groups: Group-1 consists of coefficients DC to AC-15, Group-2 consists of AC-16 to AC-31, and so on.

The 2D DCT features are used to derive a procedure for compensating the errors due to voltage overscaling in the datapath. Our procedure consists of 2 steps.

Step 1 We detect and correct errors in sign extension bits. If Table 1 specifies that a k-bit representation is sufficient, then by definition, the sign extension bits k to MSB should be all zero for a positive number and all one for a negative number. We pick three bits from the sign extension bits and used majority logic to correct the erroneous sign extension bits. This step is applicable to the groups that can be represented using 7 bits or less. False detection probability of this scheme is $C_2^3(BER_s)^2(1-BER_s)+(BER_s)^3$, where BER_s represents error rate probability of a single bit.

Step 2 We detect and correct an error when we find an abnormal increase in magnitude in one of the coefficients. This is motivated by the fact that coefficients that are adjacent to each other have similar magnitudes. The procedure is as follows. In order to detect an error in the j^{th} AC coefficient of the k^{th} block, we take the average of the two adjacent coefficients, namely, $(j-1)^{\text{th}}$ and $(j+1)^{\text{th}}$ coefficient, and compare it with the j^{th} coefficient. If the difference is higher than a predetermined threshold, we calculate the average of the j^{th} AC coefficient of the $(k-1)^{\text{th}}$ and $(k+1)^{\text{th}}$ block and compare again with the j^{th} coefficient. If the difference is again higher than the threshold, we change the value of the j^{th} coefficient to the average of the two neighboring coefficients in the same block. The pseudo code for this step is given in Algorithm 1. Since each group specified in Table 1 has different bit width specifications, we assign different threshold levels for each group to reduce the false detection probability. For instance, the threshold value for Group-1 is 64 whereas it is only 8 for Group-4. These threshold values were determined by experimentation with a sample set of images.

Algorithm 1 Pseudo Code for *Step-2*

```
Initialize Parameter Thresholds (THR_i) i = 1, 2, 3, 4
for each AC coefficient do
  if |AC(k,j) − (AC(k,j+1)+AC(k,j−1))/2| > THR_i then
    if |AC(k,j) − (AC(k+1,j)+AC(k−1,j))/2| > THR_i then
      AC(k,j) = (AC(k+1,j)+AC(k−1,j))/2
    end if
  end if
end for
```

5 Simulation Results

In this section we describe the algorithm quality performance and the hardware overhead of the two power saving schemes. The quality performance is described in terms of $PSNR$. The compression rate is measured in number of bits required to represent one pixel (bpp) and is related to the quality metric (Q). For an image

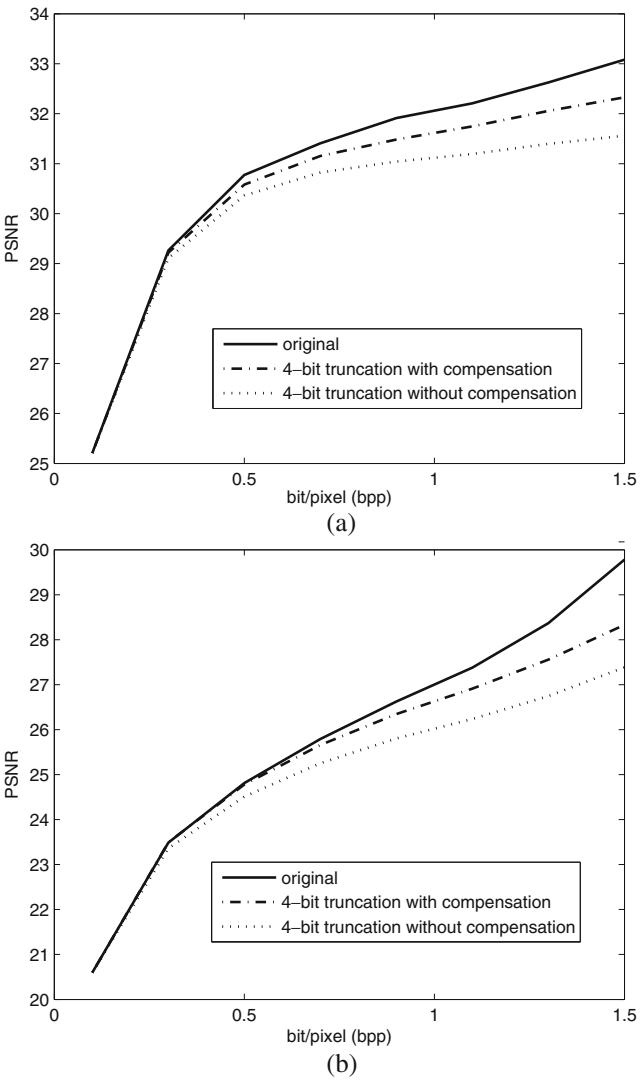


Figure 12 Performance of 4-bit truncation methods with and without compensation for (**a**) Flight and (**b**) Baboon images.

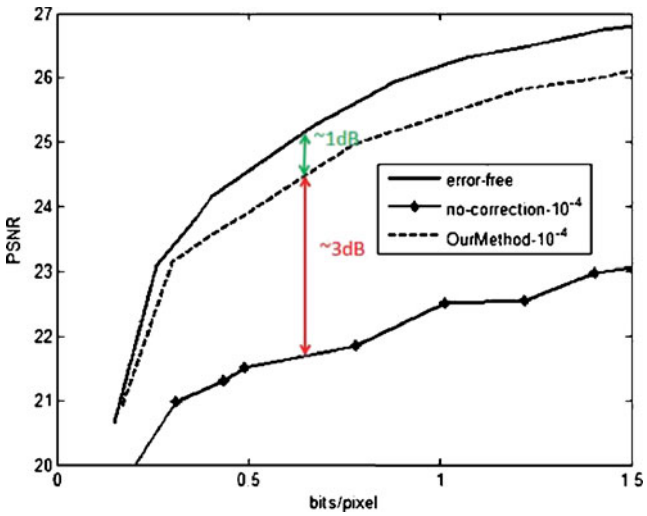
Table 2 Quality, power and latency of DCT engine for different levels of truncation.

Schemes	ΔPSNR (dB)	Active power (mW)	Latency (ns)
Baseline	0	5.39	2.92
0-bit Truncation	0	5.51	3.31
2-bit Truncation	0.1	4.76	2.95
4-bit Truncation	0.6	4.14	2.78
6-bit Truncation	2.4	3.49	2.51

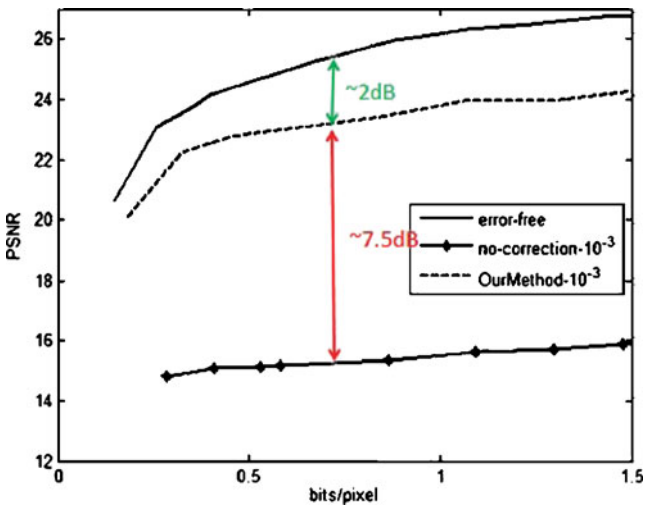
of size M by N, $I(i, j)$ is the original pixel value at (i, j) and $K(i, j)$ is the pixel value at that location after compression and decompression. If MAX_I is the maximum possible pixel value of the image, then $PSNR$ is given by Eq. 12.

$$MSE = \frac{1}{NM}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}[I(i, j) - K(i, j)]^2$$

$$PSNR = 10\log_{10}\frac{MAX_I^2}{MSE} \tag{12}$$



(a)



(b)

Figure 13 PSNR vs. compression rate performance for Bridge image when (**a**) BER(VOS) = 10^{-4} and (**b**) BER(VOS) = 10^{-3}.

Table 3 PSNR values of proposed technique at 0.75 bpp compression rate when BER(VOS) = 10^{-4}.

Images	Error free	No-correction	Proposed scheme
Bridge	25.2	21.4	24.1
Baboon	25.7	20.6	24.3
Lena	32.8	27.8	31.2
Pepper	31.5	26.4	30.3

Active power, and latency estimations of the DCT and additional circuitries are obtained using Design Compiler from Synopsys (http://www.synopsys.com) and Nangate low-power 45 nm PDK libraries [14].

5.1 Truncation Noise Compensation Method

Algorithm Performance Figure 12 illustrates the PSNR performance improvement when unbiased estimators are used for w_0 and w_1 to compensate for 4-bit truncation. For both Flight and Baboon images, the improvement is quite significant. For 1bpp ($Q \sim 50$), we observe approximately 1dB improvement compared to the system without compensation. As the number of truncation bits increases, we observe higher performance improvements using this technique.

Hardware Overhead The hardware overhead of the proposed scheme consists of two adders at the output of w_0 and w_1 units to compensate for the truncation noise, AND gates at the inputs of all the units to implement bit truncation and the associated control circuitry. Table 2 lists the power consumption and latency of the 1D DCT engine with clock period of 4 ns. The 0-bit truncation scheme includes the overhead circuitry for supporting multi-bit truncation and thus has higher power and latency compared to the baseline scheme. The active power decreases significantly with the

Table 4 Power consumption and latency of the three units in the voltage overscaling compensation scheme.

	Majority voter	Coefficient comparator	Average calculator
Active power (uW)	3.6	96	103
Latency (ps)	42	459	421

Table 5 Power consumption and ΔPSNR for various combinations of voltage scaling and low order bit truncation for a 2D DCT implementation.

Schemes	Error free		Voltage scaling with no compensation				Voltage scaling with compensation			
BER(VOS)=	0		10^{-4}		10^{-3}		10^{-4}		10^{-3}	
	ΔPSNR (dB)	Power (mW)	ΔPSNR (dB)	Power (mW)	ΔPSNR (dB)	Power (mW)	ΔPSNR (dB)	Power (mW)	ΔPSNR (dB)	Powers (mW)
0-bit Trunc	0	11.0	4.9	7.6	10.3	6.2	1.3	8.6	2.2	7.3
2-bit Trunc	0.1	9.5	4.9	6.1	10.3	5.1	1.4	7.3	2.2	6.3
4-bit Trunc	0.6	8.3	5.6	5.0	10.7	4.1	1.8	6.2	2.7	5.1
6-bit Trunc	2.4	7.0	7.1	3.8	12.3	3.3	3.5	5.0	4.2	4.5

increase in the number of truncation bits. Specifically, we see a 23% reduction in active power compared to the baseline scheme for 4-bit truncation and 35% reduction in active power for a 6-bit truncation. Table 2 also lists the change in PSNR calculated at 1 bpp ($Q \sim 50$) using 6 sample images namely, Lena, Pepper, Bridge, Baboon, Flight and House.

5.2 Voltage Scaling Compensation Method

Algorithm Performance The performance of the proposed algorithm-specific method when BER(VOS)= 10^{-4} and 10^{-3} are shown in Fig. 13 for the Bridge image using full-precision DCT. From Fig. 10, we see that when there is no truncation, 0.83 V operation results in a BER(VOS) of 10^{-4} and 0.75 V operation results in a BER(VOS) of 10^{-3}.

At BER(VOS) of 10^{-4}, our method has 3 dB improvement over the no-correction case and a drop of approximately 1 dB compared to the error-free case at 0.75 bpp compression rate ($Q \sim 30$). At BER(VOS) of 10^{-3}, quality degradation due to errors is very high as shown in Fig. 13(b). However the proposed technique helps improve the PSNR by approximately 7.5 dB at 0.75 bpp. Table 3 summarizes the performance of the proposed technique for 4 representative images (Bridge, Baboon, Lena and Pepper) at compression rate of 0.75 bpp when BER(VOS) is 10^{-4} corresponding to operating voltage of 0.83 V.

Hardware Overhead The hardware overhead of the proposed algorithm-specific consists of majority voter, coefficient comparator and average calculator. Majority voter scheme is used in the first step to detect errors in the sign extension of bits. Coefficient comparator is used to detect abnormality in magnitudes of neighboring coefficients. Average calculator is used to compensate an error bit which is rarely activated due to small number of failures. Table 4 illustrates the power consumption and latency results of the three units for clock period of 4ns. We see that the overhead is fairly small, approximately 12% of full precision 2D-DCT. Thus the proposed method enables operating at scaled voltage levels with small loss in image quality.

5.3 Combination Method

In this section we study the joint usage of bit truncation and voltage scaling techniques to further improve the power savings. The bit truncation technique not only achieves power saving but also reduces the critical path and provides extra timing slack for voltage scaling. Table 5 lists power consumption of the DCT unit and Δ PSNR for various combinations of voltage scaling and low order bit truncation for a 2D DCT implementation. Baseline scheme represents the original DCT implementation without any modification. Four truncation schemes are considered corresponding to truncation of 0-bits, 2-bits, 4-bits and 6-bits. The area of all four schemes is the same. Three scenarios for voltage scaling are considered, namely, error-free corresponding to nominal voltage operation, voltage scaling with no compensation and voltage scaling with compensation. Under voltage scaling, BER(VOS) of 10^{-4} and 10^{-3} are considered.

Sole usage of bit truncation achieves 13% to 35% reduction in power while incurring 0.1 dB to 2.4 dB PSNR degradation. When combined with voltage scaling, higher power savings of 24% to 59% is achieved while incurring 1.3 dB to 4.2 dB PSNR reduction. The voltage scaling compensation techniques are very effective in reducing ΔPSNR with only a small power overhead. For instance, for 2-bit truncation with BER(VOS)= 10^{-4}, the proposed scheme results in a 3.5 dB improvement in PSNR with only 18% increase in power consumption. Also, for the same power consumption, voltage scaling with compensation results in significant improvement in PSNR. For instance, for BER(VOS)= 10^{-4}, 4-bit truncation with voltage scaling compensation and 2-bit truncation without voltage scaling compensation have almost the same power consumption but the method with compensation has close to 3dB improvement in PSNR.

6 Conclusion

In this paper, we studied the use of bit truncation and voltage overscaling to reduce power consumption while minimizing quality degradation in JPEG codecs. The errors due to bit truncation and voltage overscaling are characterized and low overhead methods to compensate for most of these errors presented. The effect of truncation errors is minimized by using unbiased estimators. This is quite effective and simulation results show that for 4-bit truncation, this scheme achieves 23% power saving with only 0.6 dB drop in PSNR. Voltage overscaling induced errors are minimized using algorithm-specific techniques which exploit the characteristics of the quantized DCT coefficients. Operating at 0.83 V (instead of the nominal 1 V) results in a 20% reduction in datapath power but causes BER(VOS) of 10^{-4}. The proposed technique improves PSNR performance by approximately 3.4 dB compared to the no-correction case but has a degradation of about 1.2 dB in PSNR compared to the error-free case. A combination of these techniques help achieve even higher power savings with moderate decrease in PSNR. For instance, operating at 0.78V with 4-bit truncation results in power reduction of 44% with a 1.8 dB drop in PSNR.

References

1. Xanthopoulos, T., & Chandrakasan, A. (2000). Low-power DCT core using adaptive bitwidth and arithmetic activity exploiting signal correlations and quantization. *IEEE Journal of Solid State Circuits, 35*(5), 740–750.
2. Park, J., Choi, J. H., & Roy, K. (2010). Dynamic bit-width adaptation in DCT: An approach to trade off image quality and computation energy. *IEEE Transactions on VLSI Systems, 18*(5), 787–793.
3. Kim, S., Mukhopadhyay, S., & Wolf, M. (2010). System level energy optimization for error-tolerant image compression. *IEEE Embedded System Letters (ESL), 2*(3), 81–84.
4. Karakonstantis, G., Banerjee, N., & Roy, K. (2010). Process-variation resilient and voltage-scalable DCT architecture for robust low-power computing. *IEEE Transactions on VLS Systems, 18*(10), 1461–1470.
5. Kim, E. P., & Shanbhag, N. R. (2010). Soft NMR: Analysis & application to DSP systems. In *ICASSP* (pp. 1494–1497).
6. Kim, S., Mukhopadhyay, S., & Wolf, W. (2009). Experimental analysis of sequence dependence on energy saving for error tolerant image processing. In *International symposium on low power electronics and design* (pp. 347–350).
7. Cho, M., Schlessman, J., Wolf, W., & Mukhopadhyay, S. (2009). Accuracy-aware SRAM: A reconfigurable low power SRAM architecture for mobile multimedia applications. In *Asia and South Pacific design automation conference* (pp. 823–828).
8. Chang, I. J., Mohapatra, D., & Roy, K. (2009). A voltage-scalable & process variation resilient hybrid SRAM architecture for MPEG-4 video processors. In *Design automation conference* (pp. 670–675).
9. Emre, Y., & Chakrabarti, C. (2011). Data-path and memory error compensation tecnhiques for low power JPEG implementation. In *International conference on acoustic, speech and signal processing* (pp. 1589–1592).
10. Acharya, T., Tsai, P.-S. (2004). *JPEG2000 standard for image compression: Concepts, algorithms and VLSI architectures*. Wiley Inter-Science.
11. The independent JPEG Group (1998). The sixth public release of independent JPEG Group's Free JPEG Software. C Source code of JPEG Encoder research 6b, ftp://ftp.uu.net/graphics/jpeg.
12. Liu, Y., Zhang, T., & Parhi, K. K. (2010). Computation error analysis in digital signal processing systems with overscaled supply voltage. *IEEE Transactions on VLSI Systems, 18*(4), 517–526.
13. Emre, Y., & Chakrabarti, C. (2010). Memory error compensation techniques for JPEG2000. In *IEEE workshop on signal processing systems* (pp. 36–41).
14. Nangate, Sunnyvale, California (2008). 45nm open cell library. http://www.nangate.com/. Accessed Nov 2008.

Yunus Emre is a PhD student at Arizona State University. His research interests include energy and quality aware multimedia systems, error control for non-volatile and volatile memories and variation tolerant design techniques for signal processing systems.

Chaitali Chakrabarti is a professor of Electrical Engineering at Arizona State University, Tempe. Her research interests are in the areas of low-power embedded systems design and algorithm-architecture co-design of signal processing, image processing, and communication systems.