A POWER EFFICIENT SYSTEM DESIGN METHODOLOGY EMPLOYING APPROXIMATE ARITHMETIC UNITS

Tuba Ayhan, Fırat Kula, Mustafa Altun

Istanbul Technical University, Faculty of Electrical and Electronics Engineering,
Electronics and Communication Engineering Department, 34469-Istanbul, Turkey
{tuba.ayhan, firat.kula, mustafa.altun}@itu.edu.tr

ABSTRACT

In this work a power efficient approximate system design methodology is introduced and its performance is demonstrated by a 2D-DCT implementation on Spartan 3 FPGA. The method is applicable to any system with arithmetic computation regardless of their architecture, because it utilizes the existing approximate arithmetic units. The novelty of the proposed method is its system analysis approach starting from the highest level and exploring through the sub-blocks down to the basic arithmetic units. It first evaluates a given system block diagram and sets the desired performance limits of each processing block to achieve the desired ultimate quality metric. Then, the arithmetic power consumption is minimized by employing the appropriate arithmetic units which are chosen by linear/non-linear programming with linear constraint solver. The tests on 2D-DCT implementation show a power reduction of 8% for a 0.01 dB PSNR loss for 128×128 images, on the average.

Index Terms— Approximate computing, approximate adders, low-power digital systems, DCT, JPEG encoder

1. INTRODUCTION

Growing demand on mobile computing brings power-related problems along. In many applications, best quality is a low-priority concern, when compared to execution speed and power-efficiency. Approximate or only sufficiently-good results become acceptable for mobile devices because many applications are erroneous by nature since the collected data is noisy or processing algorithms are not exact. Moreover, quality can as well be traded off for power saving. The trade-off between power and accuracy leads designers to find the system that minimizes the power consumption, while still maintaining a desired quality performance.

A designer may choose and implement a signal processing algorithm only accurate enough to meet the performance criteria. This implementable algorithm is usually an approximate and low-complexity processing algorithm, which functinally resembles to the original algorithm. Good approximate algorithms such as [1] for image processing and [2] for wireless communication can be found. However, the approximate correspondents cannot be adjusted to provide different power or quality levels. In other words, this type of approximate implementations are inflexible towards changing environmental conditions and user requirements, in terms of power and quality.

A systematic design approach to optimize a system for required quality performance is required. This work proposes such method, which does not require design of a new implementable approximate algorithm. Instead, the method efficiently utilizes special arithmetic blocks with various approximation levels. Approximate adder and multiplier circuits have lower power consumption than exact arithmetic circuits. For example, with decreasing supply voltage, the power consumption of the electronic circuits is reduced at the cost of reduced accuracy [3]. Another approach is gate level or transistor level synthesizing of circuits. An overview of approximate arithmetic circuits is given in [4]. This work aims to select and employ approximate units in order to optimally meet the performance criteria of the system.

By systematic utilization of approximate computing units, maximum power savings are obtained. Moreover, the proposed approximate system design methodology can be implemented with any platform. The method investigates the overall system from the highest level down to the arithmetic units to determine the sufficient output quality at each block. Therefore, the method does not increase the algorithm or architecture design time by introducing a new block diagram. In the final phase of the method, the appropriate approximate computing units are employed in the corresponding system blocks.

The paper is organized as follows. The system design methodology is introduced in Section 2. In Section 3, the method is elaborated on a 2D-DCT implementation for a JPEG encoder. The JPEG encoding application is a simple yet sufficient example to demonstrate the methodology, because the application satisfies the two motivations: i. JPEG encoding is lossy by nature so there is no need for exact computation, ii. the PSNR of the output picture can be traded-off for power saving. The demonstration and results are also given in Section 3. The paper is concluded in Section 4.

2159-3477/17 $31.00 © 2017 IEEE
DOI 10.1109/ISVLSI.2017.50

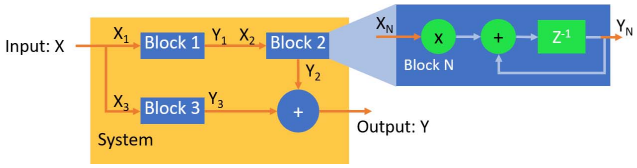

Fig. 1. A system's block diagram

2. POWER EFFICIENT APPROXIMATE SYSTEM DESIGN METHODOLOGY

A system design methodology that minimizes the total arithmetic computation power by choosing the appropriate arithmetic units among a set of available approximate and exact ones is proposed. While doing that, the designed system still maintains a certain performance which can be defined by a quality metric. The ultimate quality metric varies according to the application domain. Among many others, it can be represented as PSNR, SNR, error rate, MSE (mean squared error) ROC (receiver operating characteristic) curve, accuracy, precision etc. This method is independent from the performance representation because it links the performance with the arithmetic computation error and operates on arithmetic units.

To explain this linking method, a generic signal processing system's block diagram is given in Figure 1, where the ultimate quality performance is a function of individual blocks' performance. As we zoom into any block in the system, we realize that the blocks are composed of sub-blocks and the system components can be zoomed into computation units: multipliers and adders. This *zooming-into* operation is reflected to the design methodology as a system-to-blocks approach, which is summarized by Figure 2. The system-to-blocks approach works as follows. First, forward path is analysed to evaluate the performance limits of individual blocks. This forward analysis returns two valuable information: i. the performance metrics for blocks, ii. the relationship between sub-blocks in terms of ultimate performance goal. The first step is executed by processing a known signal such as an impulse. The output of each block is recorded and the precision of the system output is represented as a function of the block output. The purpose of this step is to convert the desired ultimate design goal of the system into error measures for blocks. As the method will be elaborated on a JPEG encoder example, the ultimate design goal of JPEG encoding can me measured by PSNR. However, the performance of its blocks such as DCT and quantization can be expressed as MSE. Moreover, each block has different contribution in the ultimate performance metric. With this step, the system is divided into blocks.

The second step is to determine the sensitivity of blocks for arithmetic calculation errors. This step works on blocks and tries to reach arithmetic sub-blocks such as adders and multipliers. The block under investigation is sensitive to the

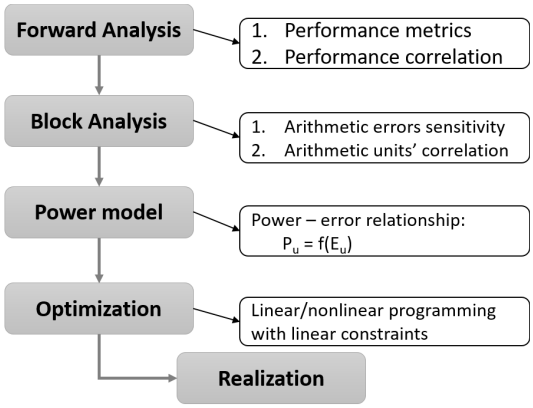

Fig. 2. System-to-blocks approach to minimize the total power consumption of arithmetic units, for a target performance

arithmetic calculation errors. In this step, both this sensitivity and the correlation between arithmetic calculation units are obtained. It is assumed that each arithmetic unit i adds zero mean white Gaussian noise with a known standard deviation σ_i to its output. A multiply-accumulate (MACC) block of the system in Figure 1 is investigated as an example. The output SNR, SNR_o is given as

$$\text{SNR}_o = \sqrt{n}\frac{\mu_2\sigma_1^2 + \mu_1\sigma_2^2}{\sqrt{\sigma_1^2+\sigma_2^2}\sqrt{\sigma_1^2\sigma_2^2 + (\sigma_m^2+\sigma_a^2)(\sigma_1^2+\sigma_2^2)}} \quad (1)$$

where, n is the number of iterations and σ_a and σ_m are standard deviation of error added by the adder and the multiplier, respectively. In this example, in order to increase the output SNR, $(\sigma_m^2 + \sigma_a^2)$ should be decreased. In other words, depending on the target SNR, this MACC can tolerate some arithmetic error, which can be quantified by (1).

The first and the second steps focus on the error metrics, but they do not contain any information about power consumption. Therefore, the next step is organized to link the performance of the arithmetic units with their power consumption. Although power consumption of an arithmetic unit increases with its accuracy in general, the exact power-accuracy relationship may vary depending on the implementation platform. A linear or non-linear model for power consumption of approximate arithmetic units is employed in the optimization algorithm. Alternatively, exact values for the power consumption can as well be utilized, but then a brute-force optimization would be required. Therefore, using exact power consumption values in the optimization algorithm is costly. Moreover, it is unnecessary because after complete system implementation, the power consumption of the units will slightly change depending on the routing. Therefore, using the exact power consumption values instead of power consumption model does not improve the optimization.

The fourth step is combining the performance metrics and power estimations obtained in the previous steps and forming

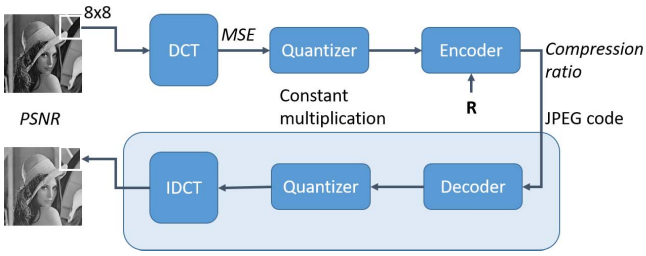

Fig. 3. JPEG encoding and decoding

an optimization problem. The total power consumption of the arithmetic units is optimized by solving a linear/non-linear programming problem with linear constraints. As stated in step 3, the optimization function -total power consumption, can be a linear or non-linear function of arithmetic unit error. The optimization problem is formulated as follows.

$$\begin{aligned} &\textbf{Minimize:} && \textstyle\sum_{i=1}^{N} P(\hat{X}_i)^2 \\ &\textbf{Subject to:} && B\hat{X} \leq b \\ & && \hat{X} \geq LB \\ & && \hat{X} \leq UB \\ & && \hat{X}_i \in \mathbb{R}\ ,\ 1 \leq i \leq N \end{aligned}$$

where, X_i is the error rate of the arithmetic unit i. $P(X)$ is the (estimated) power consumption, UB and LB are the upper and lower bounds for the error rate, respectively. LB_i is 0, if we are allowed to use exact adders and multipliers. UB_i is determined by the available approximate arithmetic units. $P(X)$ formulation, LB and UB are obtained in step 3. Constraint matrix B and target error rate b are formed according to the forward analysis in step 1 and 2. Moreover, $B\hat{X}$ should remain lower than the target error rate. The algorithm returns the optimum error rates for N arithmetic units.

The constraint matrix B is an $N \times M$ matrix and it represents the link between the M block outputs and N arithmetic units in the block. In the MACC example, $\hat{X}$ is a 2×1 vector: $\hat{X} = [e_a e_m]'$, where e_a and e_m are the additive mean errors of the approximate adder and multiplier, respectively. The adder is run for L times in the accumulator so the error of the adder will be added to the result by L times, whereas the multiplier error is included only once. Therefore, the constraint matrix for this block is $B = [L1]$.

The target error rate b is an $M \times 1$ vector. The error definitions of the block outputs must be consistent with the arithmetic unit error definitions. Therefore, an error metric conversion may be necessary, depending on the application. In the next section, an example will be demonstrated on JPEG encoder, where the ultimate error metric is PSNR and the approximate arithmetic units are classified by their mean absolute error.

3. JPEG ENCODER WITH APPROXIMATE ADDERS

The system design approach is evaluated in a JPEG encoder. Block diagram of JPEG encoding and decoding are given in Figure 3. 8×8 blocks of the image to be encoded is fed to

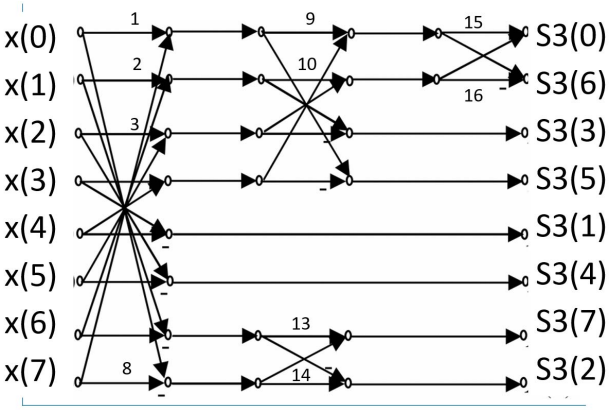

Fig. 4. 16-adder DCT signal flow diagram

2 dimensional DCT. The ouput of the DCT is quantized and encoded with entropy encoder according to the JPEG standard [5]. Image-processing related performance expectations are also written on the output of the blocks. Ultimate performance goals of a JPEG encoder are PSNR and Compression ratio. Compression ratio is adjusted by the entropy encoder; parameter **R** defines the number of AC components to be encoded. Effect of **R** on PSNR will be revealed using Lena picture after implementation in Section 3.5.

Before optimizing the blocks of the system, an architecture should be selected. For this work, a multiplierless approximate DCT architecture is selected. The details are explained below.

3.1. Multiplierless DCTs for JPEG encoding

Exact calculation of a two dimensional DCT for an 8×8 matrix g is given as

$$G_{u,v} = \alpha(u)\alpha(v) \times \sum_{x=0}^{7}\sum_{y=0}^{7} g_{x,y} cos\left(\frac{(2x+1)u\pi}{16}\right) cos\left(\frac{(2y+1)v\pi}{16}\right) \quad (2)$$

$$\alpha(x) = \begin{cases} 1/\sqrt{2} & x = 0 \\ 1 & otherwise. \end{cases} \quad (3)$$

In order to reduce complexity of DCT, this transform is converted into a matrix multiplication, in many low-complexity DCT methods. The transform with matrix multiplication is

$$\mathbf{G} = \mathbf{C} \cdot \mathbf{g} \quad (4)$$

where $\mathbf{C}$ is the transformation matrix. $\mathbf{C}$ can be approximately decomposed into a diagonal matrix $\mathbf{D}$ and a low complexity matrix $\mathbf{T}$ as

$$\mathbf{C} = \mathbf{D} \cdot \mathbf{T}. \quad (5)$$

Signed DCT (SDCT) presented in [6] arranges $\mathbf{T}$ in a way that it does not require multiplication operations nor transcendental expressions because its diagonal matrix and low

complexity matrix can be implemented with only shift operations. Bouguezel-Ahmad-Swamy Approximate DCT -will be refered as BAS-2008, [7] proposes a similar transformation, providing a higher PSNR and reduced complexity when compared to SDCT. Parametric form of BAS-2008 transform is proposed in BAS-2011 [8]. Transformation matrix of BAS-2011 contains a parameter a, whose value determines the shift operations to be processed. When the parameter a is selected as 0, then the BAS-2011 transform reaches its lowest complexity. For a is 0, the transformation is given with the low complexity matrix

$$\mathbf{T} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

and the diagonal of $\mathbf{D}$

$$[1/\sqrt{8}, 1/2, 1/2, 1/\sqrt{2}, 1/\sqrt{8}, 1/\sqrt{2}, 1/2, 1/2].$$

This transformation requires 16 adders, as the signal flow diagram shows in Figure 4.

3.2. Low power approximate JPEG encoding

Within certain limits, 16 adders of the DCT architecture in BAS-2011 can be selected from approximate adders. Therefore, $\hat{X}$ is a 16×1 vector and it returns the optimum tolerable error values for the 16 approximate adders. The relation between the adders is transformed to the constraint matrix $B_{8\times16}$, as follows:

$$B = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

The target vector b is calculated for the target PSNR as follows. First PSNR is converted into MSE using (6)

$$\text{MSE} = \frac{255^2}{10^{\left(\frac{\text{PSNR}}{10}\right)}}. \tag{6}$$

This target MSE is multiplied with the quantization matrix, in order to find the MSE of the DCT block. The constraint matrix is constructed for 1D-DCT. When it is executed for two times on the same input data, the constraints will be squared. However, the optimization problem is formed for a linear constraint matrix, so the diagonal of MSE matrix is used to determine the b vector. Since the approximate adder errors are given in percentage -as will be explained below, b vector is also converted to percentage error in [0-255] range. This way, the target percentage error of one DCT block is found.

The design method is applied in two modes: **high quality - HQ** mode and **power saving - PS** mode. The difference between the modes is the amount of tolerable noise. The HQ mode aims to minimize the power consumption for the best quality, which is at most 0.05 dB lower than the exact JPEG encoder. The PS mode on the other hand aims a power saving of at least 20%, allowing an PSNR loss of 1 dB. Choosing the PSNR values accordingly, b vector for both modes are found. Next, the adders will be introduced so that the power model inside the objective function $\sum_{i=1}^{N=16} P(\hat{X}_i)^2$ and the upper bound UB can be determined.

3.3. Approximate adders used in this work

Operation of an infinite resolution exact adder is given as $y = a + b$. If the adder is approximate, then the result of approximate addition of a and b will be $\hat{y}_{a+b}$ and the MSE of the approximate adder is defined as

$$\text{Relative MSE}_{app} = \frac{1}{(N_{max}+1)^2} \sum_{a=0}^{N_{max}} \sum_{b=0}^{N_{max}} \hat{y}_{a+b} - (a+b), \tag{7}$$

where the the maximum number that can be represented in that approximate adder is N_{max}.

Five different 13-bit approximate ripple-carry adders are used in this work. They will be abbreviated from App1 to App5, where App1 has the lowest and App5 has the highest error rate. Ripple carry adders are composed of 1-bit full adders for high order bits and approximate 1-bit adders for low-order bits. Approximate 1-bit adders are obtained by reducing the complexity of full adder: i.e. by employing an OR gate instead of an EXOR gate. The approximate adders can be varied, however design and implementation of approximate adders are beyond this work's scope. The cost of the exact adder is 40 LUT in Xilinx Spartan 6 FPGA and occupies 624 transistors in a conventional design. In Table 1, relative MSE of the approximate adders are given together with their area occupation. Logic power is proportional with the area and it also depends on the signal toggle rate. Therefore, the power consumption of these adders will change with the application, in proportion with their size in table. The optimization method and decision mechanism can be used for any implementation platform, if the power consumption estimations of discrete approximate units (adders and multipliers) are provided. Therefore, exact adder and five approximate adders are implemented on FGPA (Xilinx - xc3s500e-4cp132c) and their power consumption are plotted versus their error in Figure 5. In the figure, both a linear fit and an exponential fit are shown. Another point, an adder with very high error rate and 0 power consumption is included for the sake of curve fitting. A linear power model is used to estimate the power

	App1	App2	App3	App4	App5
Relative MSE[%]	0.0058	0.0185	0.0637	0.2737	3.8109
# FPGA LUTs	38	37	35	34	32
# Transistors	558	536	488	462	432

Table 1. Area utilization and relative MSE of approximate adders

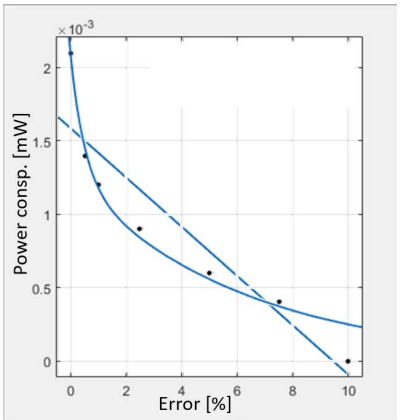

Fig. 5. Linear and exponential fits for the power consumption of different approximate adders.

consumption of the approximate adders as given in (8)

$$P = p_1 \times X + p_2 \tag{8}$$

with p_1 = -0.0001242 and p_2 = 0.0009436. So, the objective function is obtained. The lower bound LB is a zero vector, because an exact adder can be used in the design. According to the values in Table 1, the upper bound is

$$UB(i) = e_{max},\ 1 \leq i \leq N,$$

where e_{max} is 3.8109 for this set of approximate adders.

3.4. DCT design on FPGA

2D-DCT module is implemented on Xilinx Spartan 3 xcs500e-4fg320 FPGA with balanced design goal. "Keep hierarchy" mode is used in the synthesis level, in order to provide a fair comparison. The block diagram of 2D-DCT given in Figure 6 contains four blocks, as explained below.

The NxN image is processed in 8×8 sections. The **ADDRESS GENERATOR** contains a simple 3 bit counter to hold the row or column address of the 8×8 image section under processing. By the DCT blocks, BAS2011 transform matrix is implemented in a three-stage manner as illustrated in Figure 4. The DCT module is organized in a way that the 16 adders of different branches can be altered separately. **DCT1** and **DCT2** perform 1D-DCT on x and y axis, respectively. First stage of 2D-DCT is performed by **DCT1** and its output is mapped as a matrix in the **RAM BLOCK**, which is

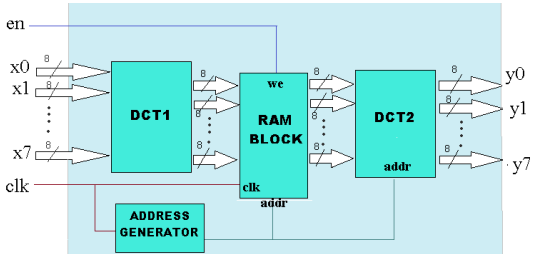

Fig. 6. 2D-DCT block diagram

64 bits by 8 bits in size. **DCT2** takes a row vector from the matrix stored in RAM block, from left to right. Its first and last input is the leftmost and the rightmost row vector value, respectively.

3.5. Implementation results

The DCT system is implemented with different adder settings. The blocks and the interconnections between the blocks are kept the same in the compared designs. The 16 adders in the DCT block are altered according to the power efficient approximate system design methodology.

Four different 128×128 images (Baboon, Barbara, Cameraman and Lena) are fed into the DCT system on FPGA. Then, transformed image is quantized and entropy encoded using Matlab. The JPEG code is then encoded on PC, using the inverse BAS-2011 transformation matrix for inverse DCT. The PSNR is calculated on the decoded image. Xilinx XPA tool is used to estimate the power consumption of the 2D-DCT system. The logic power consumption of 2D-DCT and PSNR of four 128×128 images are reported here.

For comparison, DCT FPGA implementation with three settings are used: i. with all exact adders, ii. with only one type of approximate adder and iii. with selected approximate adders according to the proposed method. Power consumption and PSNR of three settings are compared. The first setting will be named exact DCT, because exact adders are used in the implementation. In the second setting, the same approximate adder is used on each branch of the DCT module. Two approximate adders are evaluated in this setting: approximate adder 1 and 4 with a Relative MSE of 0.0058% and 0.2737%, respectively. These set ups will be called App.1 and App.4.

In the third setting, the adders are chosen among a bunch of approximate adders introduced in Section 3.3, according to the approximate system design methodology which is applied to JPEG encoding in Section 3.2.

In Figure 7, the PSNR of Lena is plotted as a function of **R**. The exact DCT and HQ approximate implementation return almost the same PSNR value, as targeted. In the PS mode, 1 dB PSNR loss was tolerated and the results show that there is an average and maximum difference of 0.82 dB and 1.18 dB is obtained, respectively. When approximate adder 1 is used at each layer, the PSNR stars to decline when com-

		Power [mW]	Power saving[%]	PSNR [dB]	% PSNR loss
App.1	Baboon	1.16	10.7692	27.7615	0.2651
	Barbara	1.12	11.1111	29.0098	0.3202
	Cameraman	1.07	12.2651	29.7574	0.2765
	Lena	1.12	12.5	26.6211	7.3253
	Average	1.1175	11.6688	28.2875	2.0468
App.4	Baboon	0.73	43.8462	23.5835	15.2748
	Barbara	0.68	46.0317	23.9752	17.6195
	Cameraman	0.57	53.2787	21.1991	28.9572
	Lena	0.65	49.2188	23.9286	16.6985
	Average	0.6575	48.0939	23.1716	19.5125
Proposed					
HQ	Baboon	1.19	8.4615	27.8282	0.0255
	Barbara	1.16	7.9365	29.0921	0.0375
	Cameraman	1.13	7.377	29.8174	0.0754
	Lena	1.18	7.8125	28.7208	0.0157
	Average	1.1650	7.8969	28.8644	0.0385
PS	Baboon	1	23.0769	26.7401	3.9346
	Barbara	0.94	25.3968	29.0543	0.1673
	Cameraman	0.9	26.2295	28.3885	4.864
	Lena	0.95	25.7813	27.6095	3.8844
	Average	0.9475	25.1211	27.9481	3.2126

Table 2. Power saving and PSNR loss due to approximate computing

pared to exact DCT result. Using approximate adder 4 drastically lowers the performance of the DCT.

The decoded image Lena is given in Figure 8. The power saving results with the implemented PSNR loss are summarized in Table 2. Let's compare the App.1 DCT with HQ DCT. Power saving of App.1 DCT is 1.5 times of HQ DCT's power saving. However, for this increase in power saving, 66 times more PSNR will be lost. A similar comparison can be made between App.4 DCT and PS DCT. Although App.4 DCT doubles the power saving provided by PS DCT, its PSNR is 6 times worse than the PS DCT's. In conclusion, selecting the approximate adders by the proposed design methodology provides more power saving than a system which is built with identical approximate arithmetic units.

4. CONCLUSION

In this paper, a power efficient approximate system design method is introduced and demonstrated on a JPEG encoder. The aim of the design methodology is to minimize the computation power, while the ultimate performance criteria of the system stays within the target limits. Regardless of the implementation platform, the method can be applied on any digital signal processing system, because it utilizes the approximate arithmetic units available for the system. The JPEG encoder which is built using this method is tested with 4 different images. As a result, 7.9% average power saving is obtained with

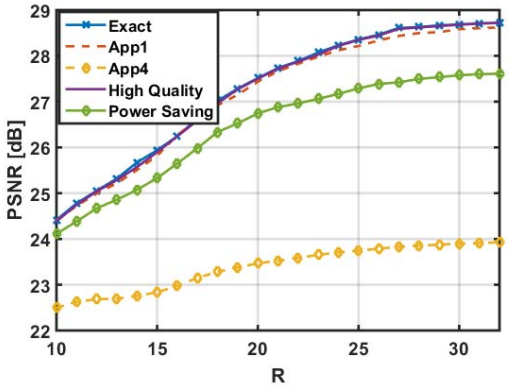

Fig. 7. Implementation results as a function of **R**

only 0.02% PSNR loss. Moreover, by changing the PSNR tolerance to 1% level, 25.12% power saving is obtained.

5. REFERENCES

[1] Jie Liang and T. D. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," *IEEE Transactions on Signal Processing*, vol. 49, no. 12, pp. 3032–3044, Dec 2001.

[2] M. Wu, B. Yin, A. Vosoughi, C. Studer, J. R. Cavallaro, and C. Dick, "Approximate matrix inversion for high-throughput data detection in the large-scale MIMO uplink," in *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, May 2013, pp. 2155–2158.

[3] D. Mohapatra, V. K. Chippa, A. Raghunathan, and K. Roy, "Design of voltage-scalable meta-functions for approximate computing," in *2011 Design, Automation Test in Europe*, pp. 1–6.

[4] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *2013 18th IEEE European Test Symposium (ETS)*, pp. 1–6.

[5] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Data Compression Standard*, Springer, 1993.

[6] Tarek I. Haweel, "A new square wave transform based on the DCT," *Signal Processing*, vol. 81, no. 11, pp. 2309–2319, 11 2001.

[7] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "Low-complexity 8x8 transform for image compression," *Electronics Letters*, vol. 44, no. 21, pp. 1249–1250.

[8] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "A low-complexity parametric transform for image compression," in *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, pp. 2145–2148.

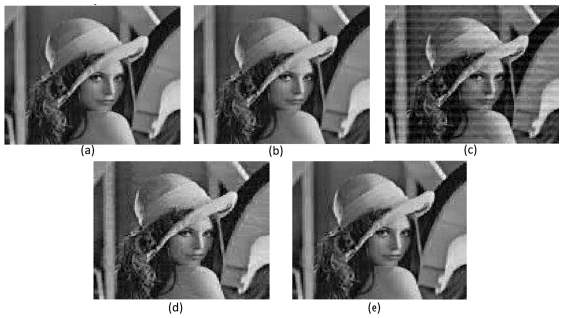

Fig. 8. Decoded lena with (a) Exact adders, (b) App.1, (c) App.4, (d) PS mode, (e) HQ mode