Dynamic Bit-Width Adaptation in DCT: An Approach to Trade Off Image Quality and Computation Energy

Jongsun Park, *Member, IEEE*, Jung Hwan Choi, *Student Member, IEEE*, and Kaushik Roy, *Fellow, IEEE*

***Abstract*—This paper presents a dynamic bit-width adaptation scheme for applications using discrete cosine transform (DCT). The technique can efficiently trade off image quality and computation energy. Based on sensitivity differences of 64 DCT coefficients, separate operand bit-widths are used for different frequency components to reduce computation energy. To select the appropriate operand bit-widths that achieve significant reduction of power consumption with minimum image quality degradation, we also propose a bit-width selection algorithm. The proposed variable bit precision DCT algorithm can be efficiently implemented using carry save adder trees. The reconfigurable DCT architecture can achieve power savings ranging from 36% to 75% compared to normal operation at the expense of minor image quality degradation.**

***Index Terms*—Discrete cosine transform (DCT), dynamic bit-width, low power design, reconfigurable architecture.**

I. Introduction

MULTIMEDIA data processing, which encompasses almost every aspect of our daily life such as communication, broadcasting, data search, advertisement, video games, etc, has become an integral part of our lifestyle. The most significant part of multimedia systems are applications involving image/video, which require computationally intensive data processing. Moreover, as the use of mobile devices increases exponentially, there is a growing demand for multimedia applications to run on these portable devices. However, multimedia applications are very computationally intensive and in lieu of limited battery capability of portable multimedia devices, energy efficient design of image/video systems is essential.

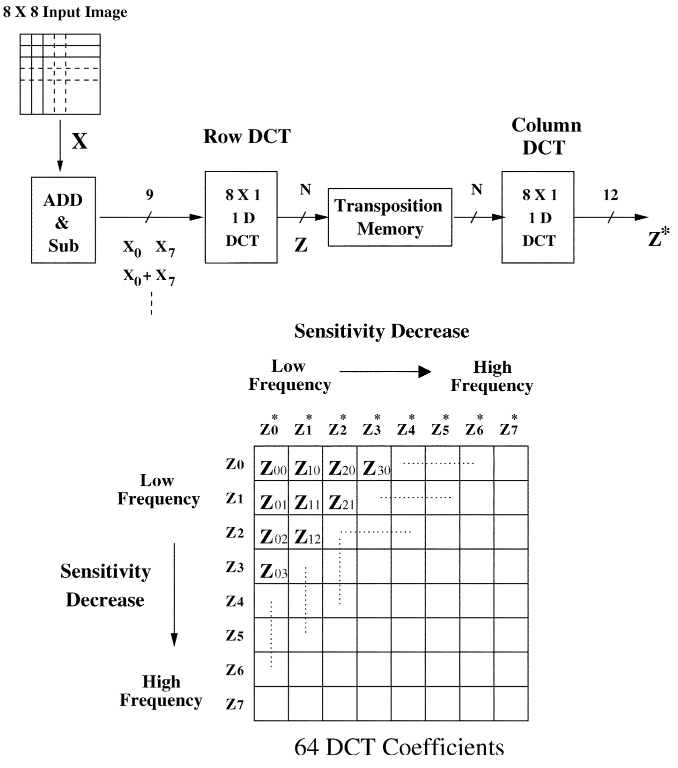


Fig. 1. 8×8 2-D DCT processor with separable 1-D DCT.

In order to reduce the volume of multimedia data over wireless channel, data compression techniques are widely used. Discrete cosine transform (DCT) [1] is one of the major compression schemes. VLSI implementation of DCT operation requires fixed-point arithmetic since floating point arithmetic needs more area and consumes higher power. In order to satisfy the power, area and system performance constraints simultaneously, accurate bit-width selection is essential.

Several techniques have been proposed to trade off image quality and DCT computation energy [2]–[5]. In [5], based on the successive approximation property of Distributed Arithmetic Operation, a scalable DCT architecture was proposed using input bit-width reduction for perceptually insignificant data. In [3], DCT bases are modified to tradeoff image quality for computational energy, while [4] proposes an efficient input bit-width adaptation scheme to reduce the power consumption.

This paper presents dynamic bit-width adaptation suitable for DCT applications to efficiently trade off image quality for lower energy of computation. Depending on the sensitivities of 64 DCT coefficients, operands of different bit-widths are used to reduce the computational complexity. To select appropriate operand bit-widths that give rise to considerable power savings with minimum image quality degradation, we propose an efficient bit-width selection algorithm. Using a circuit level technique, our variable bit precision DCT can be efficiently implemented with negligible overhead.

Manuscript received April 03, 2008; revised October 09, 2008. First published June 23, 2009; current version published April 23, 2010. This work was supported in part by Semiconductor Research Corporation and by a Korea University Grant. A portion of this paper was presented in Design, Automation, & Test in Europe (DATE) 2006, Munich, Germany.

J. Park is with the School of Electrical Engineering, Korea University, Seoul 136-701, Korea (e-mail: jongsun@korea.ac.kr).

J. H. Choi is with Digital Media and Communications R&D Center, Samsung Electronics, Suwon, Gyeounggi 443-742, Korea (e-mail: jh211.choi@samsung.com).

K. Roy is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: kaushik@ecn.purdue.edu).

Digital Object Identifier 10.1109/TVLSI.2009.2016839

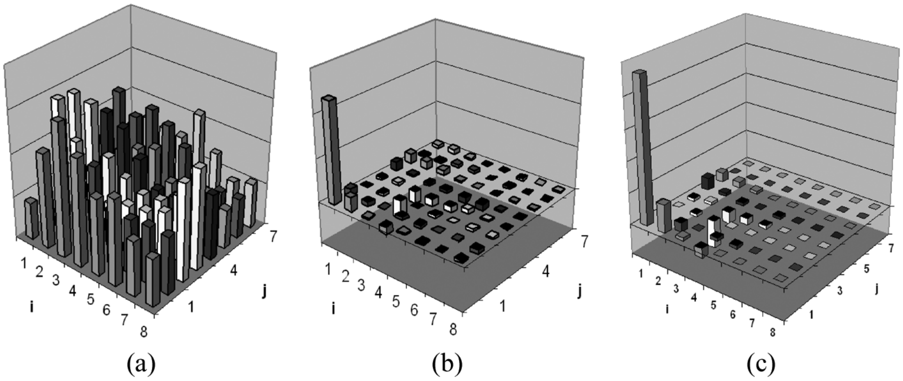


Fig. 2. (a) Normalized 8×8 block of image data. (b) Output of 2-D DCT (64 DCT coefficients). (c) Output of quantization operation.

II. DCT Architecture

In this section, we present the basic principles of DCT algorithm. In a general DCT-based source encoder [6], the original source image is partitioned into 8×8 blocks. Then, each block undergoes a 2-D DCT, whose outputs are quantized to suppress high-frequency components as human vision is less sensitive to high frequency components. The outputs from the quantizer are further compressed by an entropy encoder.

A. 2-D DCT Operation in Separable Form

The 2-D DCT process can be decomposed into an 1-D DCT (called row DCT) followed by a transposition and another 1-D DCT (called column DCT). The 1-D DCT transform is expressed in vector-matrix form as

$$z = T \cdot x^t \tag{1}$$

where T is an 8×8 matrix with DCT bases and x and z are the row and the column vectors, respectively. Since 8×8 DCT bases of matrix T in (1) has symmetric property, the 1-D DCT matrix can be rearranged and expressed as two 4×4 matrix multiplications, as shown in the following:

$$\begin{bmatrix} z_0 \\ z_2 \\ z_4 \\ z_6 \end{bmatrix} = \begin{bmatrix} d & d & d & d \\ b & f & -f & -b \\ d & -d & -d & d \\ f & -b & b & -f \end{bmatrix} \begin{bmatrix} x_0 + x_7 \\ x_1 + x_6 \\ x_2 + x_5 \\ x_3 + x_4 \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} z_1 \\ z_3 \\ z_5 \\ z_7 \end{bmatrix} = \begin{bmatrix} a & c & e & g \\ c & -g & -a & -e \\ e & -a & g & c \\ g & -e & c & -a \end{bmatrix} \begin{bmatrix} x_0 - x_7 \\ x_1 - x_6 \\ x_2 - x_5 \\ x_3 - x_4 \end{bmatrix} \tag{3}$$

where $c_k = \cos(k\pi/16)$, $a = c_1$, $b = c_2$, $c = c_3$, $d = c_4$, $e = c_5$, $f = c_6$, and $g = c_7$.

Fig. 1 shows 2-D DCT processor implemented with two 1-D DCT units and a transposition memory. In 8×1 row DCT, each column of the input data X is computed and the outputs are stored in the transformation memory. Another 8×1 column DCT is performed to obtain the desired 64 DCT coefficients.

The output bit-width N in Fig. 1 is one of the crucial parameters that determines the quality of an image. The outputs of the row DCT are truncated to N bits before they are stored in transposition memory and sent to the input of the column DCT. In our simulation, the input data bit-width of the row DCT is 9 and that of the column DCT is 12, which means $N = 12$ in Fig. 1. DCT bases in (2) and (3) are quantized with 8 bits.

B. Sensitivity to Image Quality and DCT Coefficients

The output of the 2-D DCT, which is the 8×8 block of 64 DCT coefficients, are quantized to eliminate less significant components. Each of the 64 DCT coefficients is divided by an integer number in the quantization table [6] and rounded off.

Fig. 2(a) shows an example of 8×8 block of image data in spatial domain and Fig. 2(b) shows corresponding outputs of the 2-D DCT that are 8×8 blocks of 64 DCT coefficients. Since DCT has energy compaction property, signal energy of the DCT output is concentrated on a few low frequency components while most other higher frequency components are associated with small signal energy. Fig. 2(c) shows the output data after the quantization step [6]. The visually sensitive low frequency 2-D DCT coefficients, located in the top left region of the 8×8 DCT array in the figure, are divided by small numbers while the rest of the coefficients are divided by large numbers. As a result, high frequency DCT coefficients become even smaller after quantization.

In general, the same operand bit-widths are used for calculating all the DCT coefficients. However, considering that high frequency DCT coefficients become negligibly small after quantization, we can expect that overall image quality would not be affected significantly even if we decrease the bit-width of arithmetic units used for computation of high frequency coefficients. By using smaller bit-widths for calculating high frequency components, we can achieve significant improvement in computation power at the expense of slight degradation in image quality. However, it is important to judiciously reduce the operand bit-widths for each DCT coefficient to minimize image quality degradation. In Section III, we present an algorithm to determine the appropriate bit-width configuration for DCT coefficient computation under a given image quality constraint.

III. Bit-Width Selection Algorithm in DCT

In this section, we describe a bit-width selection algorithm to efficiently tradeoff image quality and computation power.

Before going into the details of the algorithm, it is worth mentioning that we use peak signal-to-noise ratio (PSNR) as a measure of image quality. The PSNR is commonly used as a measure of the quality of reconstructed image compared to the original image [6]. It is defined as the mean squared error (MSE) of two $m \times n$ size images ORG and REC

$$\mathrm{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |\mathrm{ORG}(i,j) - \mathrm{REC}(i,j)|^2. \quad (4)$$

The PSNR is defined as

$$\mathrm{PSNR} = 20 \cdot \log_{10} \left(\frac{\mathrm{MAX}_I}{\sqrt{\mathrm{MSE}}} \right) \quad (5)$$

where MAX_I is the maximum possible pixel value of the image. Since 8 bits are used for one color component of one pixel, MAX_I is 255 in our case.

A. DCT Operand Bit-Width Selection Algorithm

The problem is formulated as follows: *Given an image and image quality constraint (in terms of PSNR), determine the operand bit-width for each DCT coefficient such that DCT computation energy is minimized.* While solving the problem, we restrict the set of permissible bit-width of adders to 12, 9, 6, 4, and 0 bits, where 0 bit means that no calculation is performed on the input. A point worth noting is that we may be able to achieve more power savings by increasing the set of permissible adder sizes, but it would significantly increase the design space to explore as well as the complexity of hardware implementation.

Fig. 3 illustrates the procedure to select a DCT operand whose bit-width is reduced. For the sake of simplicity, we use 1-D DCT as an example, but the implemented algorithm deals with the bit-widths of row DCT and column DCT operands simultaneously. In the figure, $z_0, \ldots, z_7$ are the outputs of 1-D DCT [see (2) and (3)], where z_7 stands for the highest frequency component and z_0 for the lowest frequency component. Initially, all the input bit-widths are 9 bits (the maximum input bit-width of row DCT), as shown in Fig. 3(a). First, we try to reduce the bit-width for z_7 which is the least sensitive high frequency component. We decrease the bit-width for z_7 from 9 to 6 bits and check the image quality: If it still satisfies given image constraint, this change is confirmed. In Fig. 3(b), we now have two groups: one with 9-bit width operands $(z_0 - z_6)$ and the other one with 6-bit width operand (z_7). In this case, we have two candidates for bit-width reduction: One is z_6, which is least sensitive to image quality among the first group, and the other is z_7. After calculating the PSNR of the two cases ($z_7 = 6$, $z_6 = 6$ and $z_7 = 4$, $z_6 = 9$), we select the case which gives larger PSNR and reduce the bit-width of the associated operand. Only one candidate is selected at a time and the bit-width of the selected candidate is reduced by one level (from 9 to 6 bits, from 6 to 4 bits, and so on). Figs. 3(e) and 3(f) show examples of three-candidate cases. The one with the largest PSNR among the three cases is chosen and the operand bit-width associated with that case is decreased. The algorithm continues until no candidate can satisfy the image quality constraint.

Fig. 4 shows a pseudo code for DCT operand bit-width selection algorithm. In the figure, $N_0 - N_7$ represent the input bit-widths for calculating row DCT output $z_0 - z_7$ while $N_8 - N_{15}$ are the input bit-widths for column DCT outputs $z_0^* - z_7^*$. In step 1, each N_i is initialized to the maximum bit-width (9 bits for row DCT and 12 bits for column DCT). The image quality constraint is defined in step 2. In the inner loop from steps 4 to 16, we try the candidates that are highest frequency components (least sensitive to image quality) from each of equal bit-width groups, and then we select one which leads to minimum image quality degradation. After checking the image constraint criteria, we reduce the bit-width of the chosen operand. For each iteration, DCT power savings due to input bit-width reduction are calculated and power consumption result for selected input bit-widths is updated. The iteration continues until no further input bit-width reduction is possible due to the imposed image quality constraint.

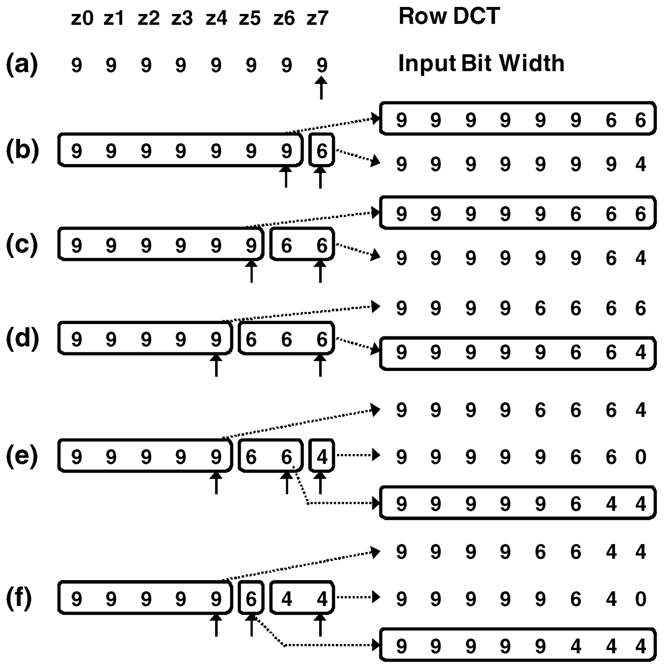


Fig. 3. Example of bit-width selection algorithm applied to row DCT.

B. Tradeoff Levels and Numerical Results

We have implemented the bit-width selection algorithm in MATLAB and considered three different cases. The image quality constraints for three cases were set to minimum allowable PSNR of 34, 32, and 30 dB, respectively, and 20 sample color images were considered. Since the algorithm presented in the previous section considers only one image, we have modified the algorithm to handle multiple images. When we select one configuration from candidates, we calculate PSNR for all the sample images (step 7 in Fig. 4) and use the lowest PSNR among them as *currentPSNR* in Fig. 4. The PSNR of compressed images are different from one image to another, thus the optimal bit-width configuration for one image may not be optimal for another image. However, many images show similar trends in image quality degradation with bit-width reduction. Therefore even though we perform bit-width selection algorithm separately for each image, common solutions can be derived in many cases.

Fig. 5 shows the input bit-width of row and column DCT for three different cases, obtained from simulation. Once the input operand bit-widths for each trade-off level is decided for hardware implementation, we do not calculate the PSNR again.

```
Pseudo code for input bit-width selection in DCT
$N_0$, $N_1$,..., $N_7$:input bit-width for $Z_0$, $Z_1$,..., $Z_7$ in row DCT
$N_8$, $N_9$,..., $N_{15}$: input bit-width for $Z_0$*, $Z_1$*,..., $Z_7$* in column DCT
BEGIN
    Initialize each Ni with the maximum precision bit width
    Define PSNR_Constraint
    LOOP
        Initialize maxPSNR to 0
        FOR each N, LOOP
            IF (Ni is least sensitive in an equal bit-width group) then
                Decrease Ni by one level
                currentPSNR = Calculate_PSNR ( $N_0$, $N_1$, $N_2$, ... $N_{15}$ )
                currentEnergy = Calculate_ENERGY ( $N_0$, $N_1$, $N_2$, ... $N_{15}$)
                IF (currentPSNR > maxPSNR) then
                    minEnergy = currentEnergy
                    maxPSNR = currentPSNR
                    optBitWidthArray = ($N_0$, $N_1$, $N_2$, $N_3$ ... $N_{15}$)
                END IF
                Restore Ni (Increase Ni by one level)
            END IF
        END FOR
        If (maxPSNR < PSNR_Constraint) then BREAK
        ( $N_0$, $N_1$, $N_2$, $N_3$ ... $N_{15}$ ) = optBitWidthArray
    END LOOP
END
```

Fig. 4. Pseudo code for input bit-width selection in DCT.

	Row DCT								Column DCT							
	z_0	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_0^*	z_1^*	z_2^*	z_3^*	z_4^*	z_5^*	z_6^*	z_7^*
Normal Operation	9	9	9	9	9	9	9	9	12	12	12	12	12	12	12	12
	z_0	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_0^*	z_1^*	z_2^*	z_3^*	z_4^*	z_5^*	z_6^*	z_7^*
Trade off case 1	9	9	6	6	6	4	0	0	12	12	9	9	9	6	0	0
	z_0	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_0^*	z_1^*	z_2^*	z_3^*	z_4^*	z_5^*	z_6^*	z_7^*
Trade off case 2	9	6	4	4	0	0	0	0	12	9	6	6	0	0	0	0
	z_0	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_0^*	z_1^*	z_2^*	z_3^*	z_4^*	z_5^*	z_6^*	z_7^*
Trade off case 3	9	4	4	0	0	0	0	0	9	6	4	0	0	0	0	0

Fig. 5. Input operand bit-width in 2-D DCT operation for three tradeoff levels.

TABLE I
PSNR VALUES FOR SEVEN COLOR IMAGES FOR DIFFERENT TRADEOFF LEVELS

PSNR(dB)	Original	Level 1	Level 2	Level 3
lena	34.97	34.85	33.79	31.51
tulips	35.80	35.66	33.94	30.84
peppers	36.16	35.55	33.02	30.60
monarch	36.05	35.88	34.00	31.08
lady	35.97	35.79	34.34	31.97
sail	34.40	34.15	32.75	30.02
fruits	35.86	35.54	33.40	30.74

In normal operation, 9-bit inputs and 12-bit inputs are used for row DCT and column DCT, respectively. As we go to the higher tradeoff levels (sacrificing image quality in favor of lower power), the input bit-width for calculating both row and column DCT is reduced, starting from the less sensitive high frequency components. Table I shows the PSNR values of 7 images for each tradeoff level.

IV. Low Power Reconfigurable DCT

In this section, we present a DCT architecture, which can be reconfigured from one tradeoff level to another without significant hardware overhead.

A. *DCT Implementation Using Carry Save Adder*

We use a carry save adder tree architecture [7] for 1-D DCT matrix multiplications [see (2) and (3)]. Let us consider the example of calculating z_1. From (3), z_1 can be expressed as

$$z_1 = a(x_0 - x_7) + c(x_1 - x_6) + e(x_2 - x_5) + g(x_3 - x_4). \quad (6)$$

DCT bases (a, c, e, g) are represented as canonical signed digits (CSD) [3] in order to reduce the number of additions for calculating z_1. Fig. 6(a) shows the implementation of (6) with a carry save adder tree. In the figure, over-lines indicate negative sign and "$\ll$" corresponds to bit-wise left shift. Total 9 adders are used to compute z_1. Other outputs of 1-D DCT operation, which are $z_0, z_2, \ldots z_7$ [see (2) and (3)], can be similarly implemented using carry save adder trees.

B. *Reconfigurable DCT Implementation*

In Section III-B, we proposed 3 different image quality / computation energy tradeoff, where the DCT input bit-widths are changed. In the previous work [3], AND gates were used to dynamically change the input bit-widths of arithmetic units by gating the inputs. However, using AND gates for input gating incurs large area overhead. In this work, we propose a simple circuit level technique to implement a reconfigurable DCT architecture with negligible area overhead.

Fig. 7(a) shows the schematic of general complementary logic gate [7]. For reconfigurability, two transistors are added to the gate: one pMOS in series with pull-up network (PUN) and one nMOS in parallel with pull-down network (PDN) as shown in Fig. 7(b). When the control signal ϕ is equal to ZERO, the

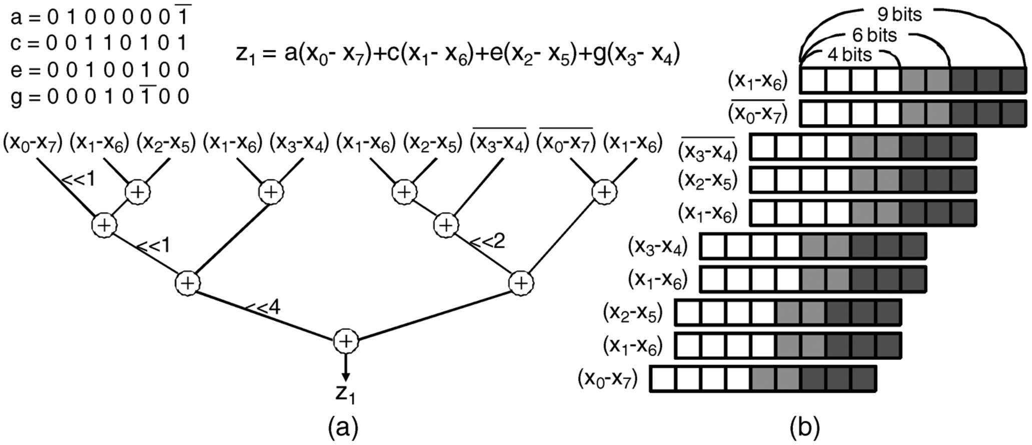


Fig. 6. (a) Balanced adder tree architecture for calculating z_1. (b) Inputs of the carry save adder tree.

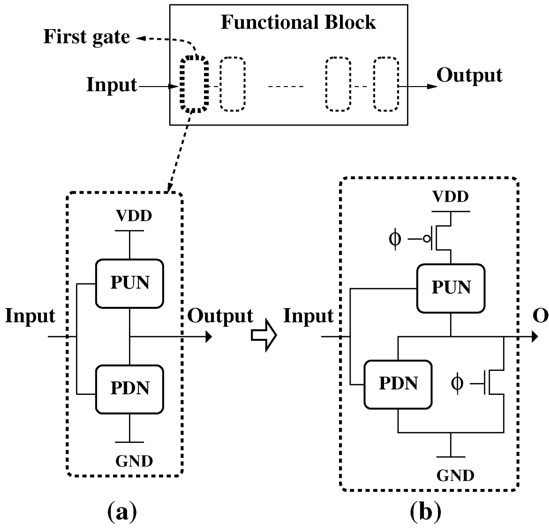


Fig. 7. Circuit level approach to turn off functional block. (a) General gate schematic. (b) Modified gate schematic to turn off functional block.

circuit works like the standard gate shown in Fig. 7(a). There is small increase in delay due to an additional pMOS transistor. When the control signal ϕ is equal to ONE, the output of the gate is forced to zero regardless of input signal. If we modify only the first gate of the functional block (see Fig. 7) in this way, assigning ONE to ϕ is equivalent to turning off the functional block since there is no switching activity in the following nodes. Using the proposed circuit level technique, area overhead for input gating can be significantly reduced. When AND gates are used for gating in the reconfigurable DCT architecture, the area overhead was 5.07%. However, the overhead reduces to 1.2% when the proposed circuit technique is used.

The proposed circuit approach is applied to the carry save adder implementation shown in Fig. 6(a). Each of the first stage logic gate in a carry save adder tree is modified to the gate with ϕ signal in Fig. 7(b). Fig. 6(b) shows the inputs of the carry save adder tree in Fig. 6(a). In normal operation, full precision bit-width (9 bits) is used. As we go to higher tradeoff level to reduce power consumption, we decrease the input bit-width to 6 or 4 bits by forcing least significant bits to zeros. As the input operand bit-width is reduced from 9 to 6 bits or from 6 to 4 bits, power saving is achieved at the expense of degradation in image quality.

TABLE II
POWER CONSUMPTION AND IMAGE QUALITY AT DIFFERENT TRADEOFF LEVELS

	Normal operation	Trade-off level 1	Trade-off level 2	Trade-off level 3
Power (mW)	94.05	60.54	37.70	23.8
Percentage	100%	63.3%	39.9%	25.2%
PSNR (dB)	36.16	35.55	33.02	30.60

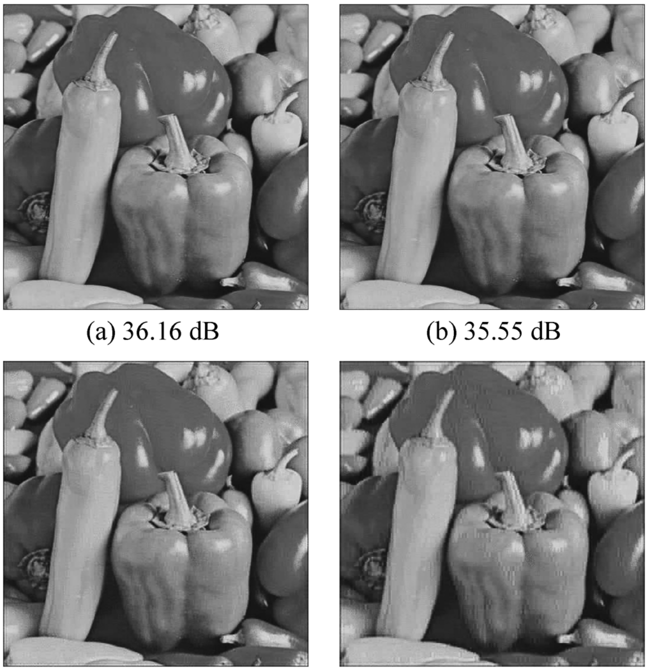
C. Power Consumption Results

The reconfigurable DCT architecture using carry save adder trees is implemented in *TSMC* 0.25-μm technology. Using the control signal ϕ in Fig. 7, the proposed DCT architecture can be dynamically reconfigured from one tradeoff level to another with negligible area overhead.

Table II shows image quality (in terms of PSNR) and power consumption for different tradeoff levels. Power consumption is measured by simulation of SPICE netlist using *nanosim* [8] and *peppers* image is used for the simulation. At tradeoff level 1, we can achieve around 36% of power savings compared to the original configuration. The PSNR decreases from 36.16dB to 35.55dB. When further power saving is required, the proposed DCT architecture can be reconfigured to tradeoff level 2 and level 3, which leads to 60% or 75% power savings, respectively. Fig. 8 shows the *peppers* image for different tradeoff levels.

In Table III, we compare power dissipation and image quality with previous approaches proposed in [3], [5] for *peppers* image. Compared to adaptive bit-width approach in distributed arithmetic based architecture in [5] and the DCT basis modification approach in [3], our proposed scheme shows larger reduction in power consumption for comparable image quality degradation.

Image quality requirement may be different from one application to another. High definition displays may require higher PSNR, while small degradation in image quality may not be



(a) 36.16 dB

(b) 35.55 dB

(c) 33.02 dB

(d) 30.60 dB

Fig. 8. (a) Original peppers image (100% power consumption, 36.16 dB). (b) In tradeoff level 1 (63.3% power consumption, 35.55 dB). (c) In level 2 (39.9% power consumption, 33.02 dB). (d) In level 3 (25.2% power consumption, 30.60 dB).

TABLE III
POWER AND IMAGE QUALITY COMPARISON WITH PREVIOUS APPROACHES

		Normal	level 1	level 2	level 3
proposed	PSNR (dB)	36.16	35.55	33.02	30.60
	power (%)	(100%)	(63.3%)	(39.9%)	(25.2%)
approach in [3]	PSNR (dB)	36.16	35.62	33.18	30.4
	power (%)	(100%)	(80.4%)	(53.8%)	(31.7%)
approach in [5]	PSNR (dB)	36.0	35.5	33.0	30.0
	power (%)	(100%)	(98.0%)	(93.0%)	(86.0%)

visible on small screens of mobile systems. The proposed technique is more suitable for mobile applications where longer battery life is favored over high image quality. When the battery strength is low in a mobile system, the user or system can choose to trade off image quality for power consumption to extend the battery life. Depending on the required amount of power saving, the proposed scheme allows the selection of different bit-width configuration in DCT architectures, thus achieving considerable reduction in power consumption at the expense of image quality degradation.

V. CONCLUSION

In this work, we propose a low power, reconfigurable DCT architecture to allow efficient tradeoff between image quality and computation energy. The DCT architecture uses the dynamic bit-width adaptation, where operand bit-widths are changed according to image quality and/or power consumption requirements. Different tradeoff levels are specified and the proposed DCT architecture can be dynamically reconfigured from one tradeoff level to another. The proposed reconfigurable DCT architecture leads to 36% power savings with little degradation in image quality (0.61 dB). With the proposed architecture, larger computation power savings can be achieved at the expense of additional degradation in image quality.

REFERENCES

[1] N. Ahmed, T. Natarajan, and K. R. Rao, “Discrete cosine transform,” *IEEE Trans. Computer*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.

[2] A. Sinha, A. Wang, and A. Chandrakasan, “Energy scalable system design,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 2, pp. 135–145, Apr. 2002.

[3] J. Park and K. Roy, “A low power reconfigurable DCT architecture to trade off image quality for computational complexity,” in *Proc. Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2004, vol. 5, pp. 17–20.

[4] J. Park, J. Choi, and K. Roy, “Dynamic bit-width adaptation in DCT: image quality versus computation energy trade-off,” in *Proc. Des., Autom., Test Eur. (DATE)*, Mar. 2006, pp. 520–521.

[5] T. Xanthopoulos and A. Chandrakasan, “A low-power DCT core using adaptive bitwidth and arithmetic activity exploring signal correlations and quantization,” *IEEE J. Solid-State Circuits*, vol. 35, no. 5, pp. 740–750, May 2000.

[6] *Digital Compression and Coding of Continuous-Tone Still Images*, ITU-T Recommendation T.81, Sep. 1992.

[7] J. M. Rabaey, A. Chandrakasan, and B. Nikolic*, Digital Integrated Circuits*, 2nd ed. Englewood Cliffs, NJ: Prentice–Hall, 2003.

[8] Synopsys Inc., San Jose, CA, “Nanosim reference guide,” Jun. 2001.

Jongsun Park (M’05) received the B.S. degree in electronics engineering from Korea University, Seoul, Korea, in 1998 and the M.S. and Ph.D. degrees in electrical and computer engineering from Purdue University, West Lafayette, IN, in 2000 and 2005, respectively.

He joined the Electrical Engineering faculty of the Korea University, Seoul, Korea, in 2008. From 2005 to 2008, he was with the Signal Processing Technology Group, Marvell Semiconductor Inc., Santa Clara, CA. He was also with the Digital Radio Processor System Design Group, Texas Instruments, Dallas, TX, in Summer of 2002. His research interests focus on variation-tolerant, low-power and high-performance VLSI architectures and circuit designs for digital signal processing and digital communications.

Jung Hwan Choi (S’07) received the B.S. and M.S. degrees in electrical engineering from Seoul National University, Seoul, Korea, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, in 2009.

Presently, he is a Senior Engineer with Samsung Electronics, Korea. In the summer of 2006 and 2007, he was with Intel Corporation, Austin, TX, as an intern. His research interests include low-power DSP circuit design, power-aware system-on-chip (SoC) design methodologies, and thermal modeling and analysis.

Kaushik Roy (SM'95–F'01) received the B.Tech. degree in electronics and electrical communications engineering from the Indian Institute of Technology, Kharagpur, India, and the Ph.D. degree from the Electrical and Computer Engineering Department, University of Illinois at Urbana-Champaign, Urbana-Champaign, in 1990.

He was with the Semiconductor Process and Design Center, Texas Instruments, Dallas, TX, where he worked on FPGA architecture development and low-power circuit design. He joined the electrical and computer engineering faculty at Purdue University, West Lafayette, IN, in 1993, where he is currently a Professor and holds the Roscoe H. George Chair of Electrical and Computer Engineering. His research interests include VLSI design/CAD for nano-scale Silicon and non-Silicon technologies, low-power electronics for portable computing and wireless communications, VLSI testing and verification, and reconfigurable computing. He has published more than 450 papers in refereed journals and conferences, holds 8 patents, and is coauthor of two books on low power CMOS VLSI design.

Dr. Roy was a recipient of the National Science Foundation Career Development Award in 1995, IBM Faculty Partnership Award, ATT/Lucent Foundation award, 2005 SRC Technical Excellence Award, SRC Inventors Award, Purdue College of Engineering Research Excellence Award, and Best Paper Awards at 1997 International Test Conference, IEEE 2000 International Symposium on Quality of IC Design, 2003 IEEE Latin American Test Workshop, 2003 IEEE Nano, 2004 IEEE International Conference on Computer Design, 2006 IEEE/ACM International Symposium on Low Power Electronics and Design, and 2005 IEEE Circuits and System Society Outstanding Young Author Award (Chris Kim), 2006 IEEE Transactions on VLSI Systems Best Paper Award. He is a Purdue University Faculty Scholar. He was a Research Visionary Board Member of Motorola Labs (2002). He has been in the editorial board of IEEE DESIGN AND TEST, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, and IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. He was Guest Editor for a Special Issue on Low-Power VLSI in the IEEE DESIGN AND TEST (1994) and IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS (June 2000), *IEE Proceedings—Computers and Digital Techniques* (July 2002).