A Low-Power DCT Core Using Adaptive Bitwidth and Arithmetic Activity Exploiting Signal Correlations and Quantization

Thucydides Xanthopoulos, *Student Member, IEEE*, and Anantha P. Chandrakasan, *Member, IEEE*

***Abstract*—This work describes the implementation of a discrete cosine transform (DCT) core compression system targetted to low-power video (MPEG2 MP@ML) and still-image (JPEG) applications. It exhibits two innovative techniques for arithmetic operation reduction in the DCT computation context along with standard voltage scaling techniques such as pipelining and parallelism. The first method dynamically minimizes the bitwidth of arithmetic operations in the presence of data spatial correlation. The second method trades off power dissipation and image compression quality (arithmetic precision.) The chip dissipates 4.38 mW at 14 MHz and 1.56 V.**

***Index Terms*—Discrete cosine transform, low power DSP, video compression.**

I. Introduction

RECENTLY, there has been significant interest in portable electronic devices with digital signal-processing (DSP) capabilities. Applications include portable video, wireless communications, and sensor data processing. This paper focuses on two novel aspects of low-power DSP system design: Exploitation of correlation and reduced data dynamic range for switching activity reduction and precision reduction for computation related to perceptually insignificant data.

The case study presented in this paper is a low-power discrete cosine transform (DCT) core processor with direct applications to video and still-image compression systems (MPEG, JPEG). The main design ideas can be adapted to other DSP contexts where the input data exhibit substantial correlation or reduced dynamic range. Two innovative techniques have been developed for arithmetic activity reduction. The first method exploits the fact that image pixels are locally well correlated and exhibit a certain number of common most significant bits (MSB's). These bits constitute a common-mode dc offset that only affects the computation of the dc DCT coefficient and is irrelevant for the computation of the higher spectral coefficients. This observation follows directly from the linearity of the transform. The DCT chip uses adaptive-bitwidth distributed arithmetic (DA) computation units that reject common MSB's for all ac coefficient computations, resulting in arithmetic operations with reduced bitwidth operands and thus reducing switching activity. We call this method MSB rejection (MSBR).

The second method exploits the fact that in image and video compression applications, the DCT is always followed by a quantization step, which essentially reduces the precision of the visually insignificant higher spatial frequencies. The DCT chip allows the user to program statically the maximum desired precision of each spectral coefficient on a row-by-row basis so that no unnecessary computation is performed if the precision will be lost anyway due to quantization. Additional dynamic adaptation in computation precision is provided by a row–column peak-to-peak detector that classifies each block row and column into one of four classes of computation precision for maximizing image peak signal-to-noise ratio (PSNR) and minimizing the number of arithmetic operations. We call this method row–column classification (RCC). The chip user can define precisely all four classes in addition to the desired precision per coefficient per class.

II. Background

Past researchers have designed DSP subsystems that take advantage of reduced dynamic range and reduced precision tolerance. Nielsen and Sparso [1], [2] have proposed a sliced data path for a digital hearing-aid filter bank that exploits small magnitude arithmetic inputs for low power. The arithmetic data path has been partitioned in an MSB and a least significant bit (LSB) slice. The MSB slice is only enabled when the input bitwidth requires it. Activation of the slices is performed by using special data tags that indicate the presence of sign extension bits in the MSB input slice. Additional circuit overhead is required for the computation and update of tags. Moreover, dynamic bitwidth adaptation is very coarse and can only be performed on a slice basis.

Ludwig *et al.*[3] have demonstrated an approximate filtering technique, which dynamically reduces the filter order based on the input data characteristics. More specifically, the number of taps of a frequency-selective finite-impulse response filter is dynamically varied based on the estimated stopband energy of the input signal. The resulting stopband energy of the output signal is always kept under a predefined threshold. This technique results in power savings of a factor of six for speech inputs. Larsson and Nikol [4], [5] have demonstrated an an adaptive scheme for dynamically reducing the input amplitude of a Booth-encoded multiplier to the lowest acceptable precision

Manuscript received July 29, 1999; revised November 20, 1999. This work was supported in part by the Defense Advanced Research Projects Agency and in part by National Semiconductor Corporation under a graduate fellowship.

The authors are with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

Publisher Item Identifier S 0018-9200(00)02986-3.

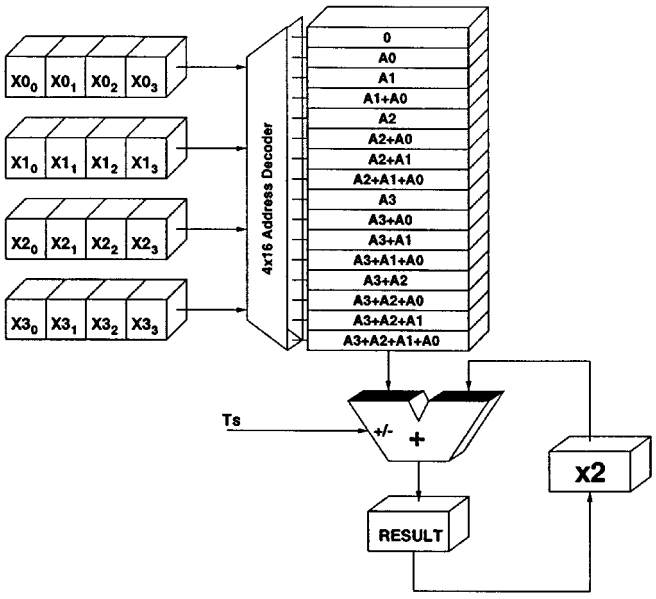

Fig. 1. Distributed arithmetic ROM and accumulator structure.

level in an adaptive digital equalizer. Their scheme simply involves an arithmetic shift (multiplication/division by a power of two) of the multiplier input depending on the value of the error at the equalizer output. They report power savings of 20%.

The present chip will demonstrate arithmetic units that can be easily adapted to reject sign extension bits at a very fine granularity and can dynamically vary their bitwidth at runtime. All arithmetic operations in the core processor are executed by distributed arithmetic units, which do not require the use of a multiplier. The distributed arithmetic computation mechanization is a very important component of the system, and its properties are heavily exploited for power reduction.

DA [6], [7] is a bit-serial operation that computes the inner product of two vectors (one of which is a constant) in parallel. Its main advantage is the efficiency of mechanization and the fact that no multiply operations are necessary. DA has an inherent bit-serial nature, but this disadvantage can be completely hidden if the number of bits in each variable vector coefficient is equal or similar to the number of elements in each vector.

Fig. 1 shows a detailed example of a distributed arithmetic computation. The structure shown computes the dot product of a four-element vector X and a constant vector A. All 16 possible linear combinations of the constant vector elements (A_i) are stored in a ROM. The variable vector X is repackaged to form the ROM address most significant bit first. We have assumed that the X_i elements are 4-bit 2's-complement integers (bit 3 is the sign bit.) Every clock cycle the RESULT register adds $2\times$ its previous value (reset to zero) to the current ROM contents. Moreover, during each cycle, the four registers that hold the four elements of the X vector are shifted to the right. The sign timing pulse T_s is activated when the ROM is addressed by bit 3 of the vector elements (sign). In this case, the adder subtracts the current ROM contents from the accumulator state. After four cycles (bitwidth of the X_i elements), the dot product has been produced within the RESULT register.

Since the DA mechanization is essentially a bit-serial dot product operation, it exhibits successive approximation properties if it is performed MSB first. Each successive intermediate dot product value within the RESULT register is closer to the final value in a stochastic sense. A detailed derivation can be found in [8].

III. DCT Algorithm and Chip Architecture

The DCT processor implements a row–column distributed arithmetic version of the Chen [9] fast DCT algorithm enhanced with the activity-reduction methods outlined in Section I, namely, MSB rejection and row–column classification.

The first step of the Chen algorithm is a factorization of the DCT-II [10] matrix such that the subsequent computation of the even coefficients is fully separated from the computation of the odd coefficients. The 8-point one-dimensional (1-D) DCT can be expressed as follows:

$$\begin{bmatrix} X_0 \\ X_2 \\ X_4 \\ X_6 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} A & A & A & A \\ B & C & -C & -B \\ A & -A & -A & A \\ C & -B & B & -C \end{bmatrix} \cdot \begin{bmatrix} x_0 + x_7 \\ x_1 + x_6 \\ x_2 + x_5 \\ x_3 + x_4 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} X_1 \\ X_3 \\ X_5 \\ X_7 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} D & E & F & G \\ E & -G & -D & -F \\ F & -D & G & E \\ G & -F & E & -D \end{bmatrix} \cdot \begin{bmatrix} x_0 - x_7 \\ x_1 - x_6 \\ x_2 - x_5 \\ x_3 - x_4 \end{bmatrix} \quad (2)$$

where

$$A = \cos\frac{\pi}{4}, \quad B = \cos\frac{\pi}{8}, \quad C = \sin\frac{\pi}{8}, \quad D = \cos\frac{\pi}{16},$$
$$E = \cos\frac{3\pi}{16}, \quad F = \sin\frac{3\pi}{16}, \quad G = \sin\frac{\pi}{16}. \quad (3)$$

Fig. 2 shows a block diagram of the system architecture. It consists of two 8-point 1-D DCT stages with a transposition memory structure (TRAM) in between. The TRAM design is described in [11].

We first concentrate on the basic functionality of this subsystem, ignoring for the moment the *row and column classifiers* that reduce the computation precision and the control blocks named *MSB rejection* that implement a sliding computation window that rejects common most significant bits and sign extension bits for the core computations.

Input pels are shifted into registers D0–D7 of the first stage at a rate of one sample per clock. The subsequent column of adders and subtractors perform the first "butterfly" step contained in the last column vectors of (1) and (2). Data are then loaded into eight shift registers that repackage the data into two 4-bit addresses that serially feed the ROM and accumulators (RAC0–RAC7) MSB first. Each ROM and accumulator structure (RAC) is a distributed arithmetic structure (Fig. 1) that implements the dot product of the four-element even (odd) processed pel vector and a row of the 4×4 coefficient matrix in (1) and (2). The ROM within each RAC contains linear combinations of the coefficient matrices row vector elements. Once every eight cycles (given that pel sums and differences are rounded to 8 bits), the eight dot products have assumed their final values within each RAC. The ROM within each RAC is a 16 by 10-bit-wide structure,

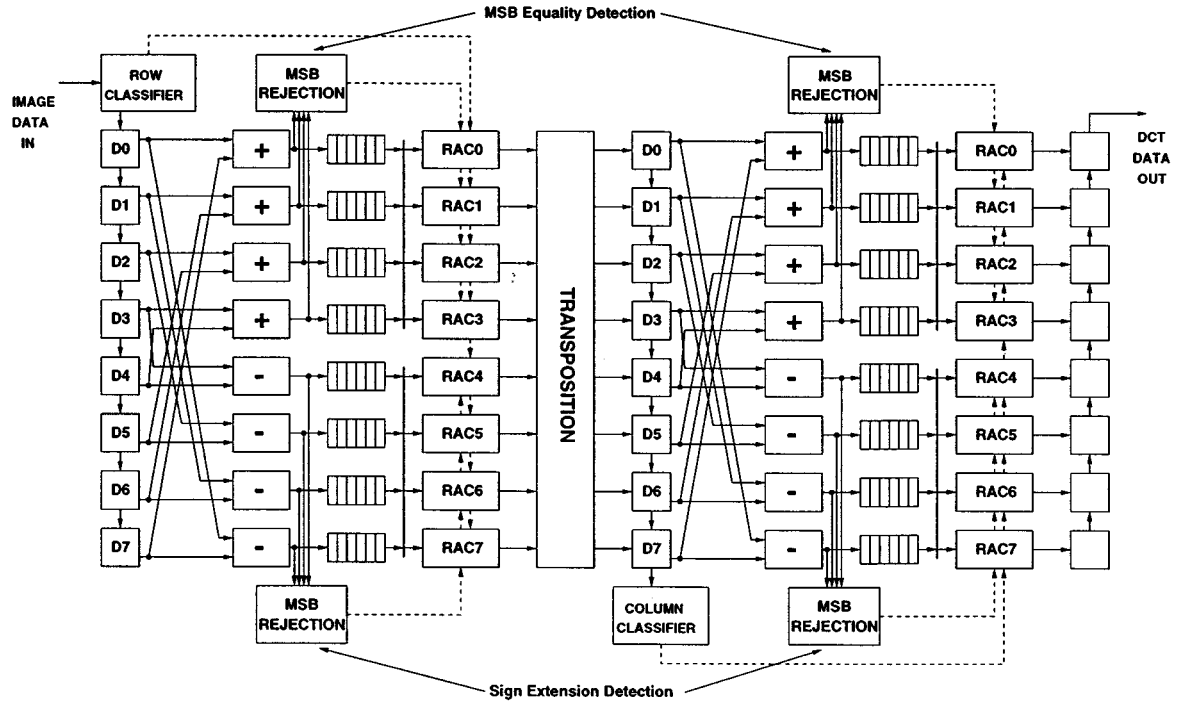

Fig. 2. DCT core processor block diagram.

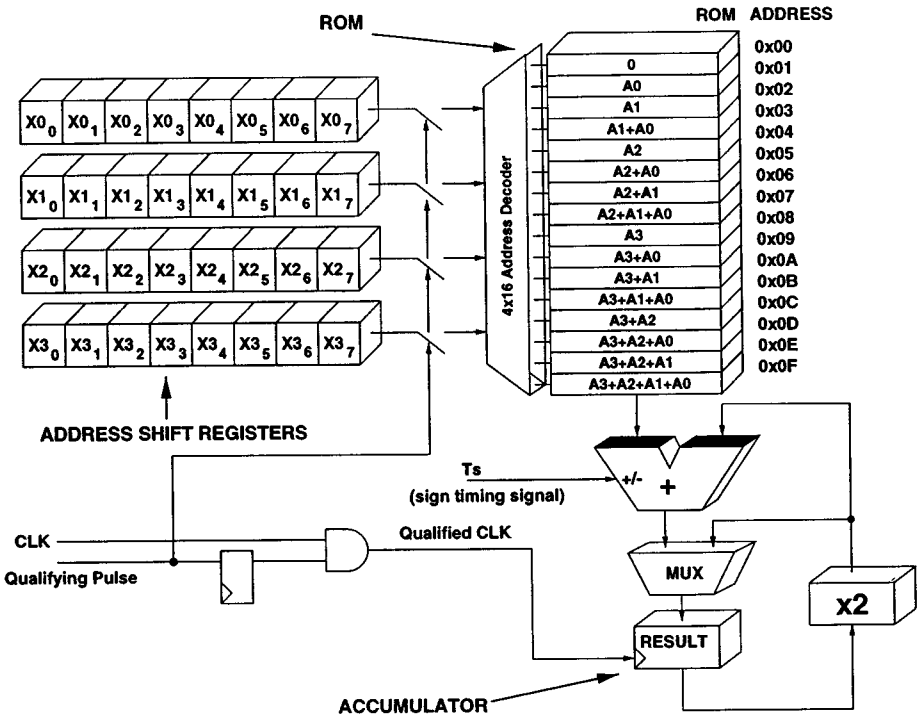

Fig. 3. DCT ROM and accumulator.

whereas the adder is 20 bits wide to accommodate continuously increasing operands due to the left logical shift in the accumulator feedback path. A detailed diagram of the DCT RAC's is shown in Fig. 3. The ROM contents are labelled as follows. If the row vector $[A0\ A1\ A2\ A3]$ is replaced with a row of either constant matrix in (1) and (2), the resulting RAC computes the vector element in the same row position as the matrix row on the left-hand side of (1) and (2). We defer the discussion of the qualifying pulse until Section III-A.

Subsequent processing involves loading the results of the eight distributed arithmetic computations in a transposition structure (first stage) or loading in an eight-wide parallel-to-serial register that will output the coefficient stream at a rate of one sample per clock (second stage.)

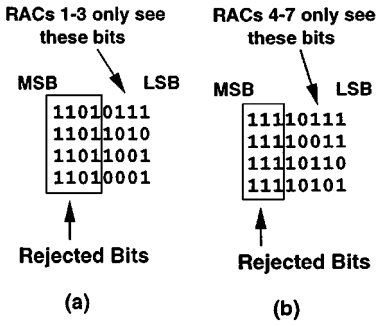

Fig. 4. MSB equality detection and sign extension detection example.

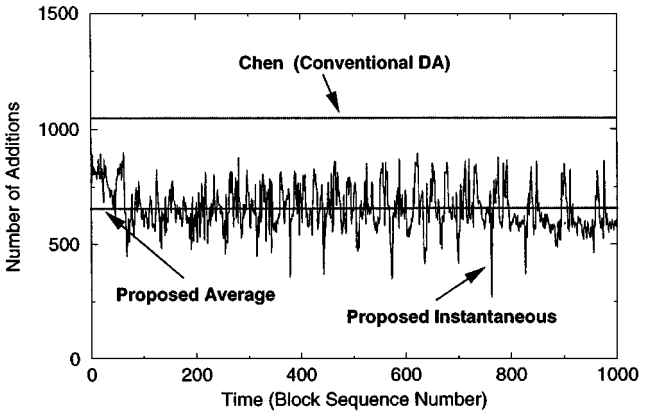

Fig. 5. Comparison of DCT additions (no RCC).

Incoming pels are 8-bit-wide unsigned integers. The result of the butterfly step is 9-bit 2's-complement and is rounded to 8 bits. The results of the first-stage eight DA computations are 20-bit 2's-complement and are rounded to 11-bit 2's-complement before being passed to the second stage. The result of the second-stage butterfly stage is 12-bit 2's-complement rounded to 11-bit 2's-complement. Second-stage RAC's 1–7 use only eight significant bits of those 11 (using a sliding computation window, as explained in Section III-A). RAC0 though uses the full 11 bits due to its high visual significance. The results of all RAC's are 20-bit 2's-complement rounded to 12-bit 2's-complement before being shifted out of the chip.

A. MSB Rejection

We now focus on the mechanics of MSB rejection. Let us focus on the top half of the block diagram (both stages) in Fig. 2. Due to spatial correlation in natural images, pel sums $x_0 + x_7, x_1 + x_6, x_2 + x_5, x_3 + x_4$ [(1)] are likely to have a number of equal most significant bits. By definition, all distributed arithmetic ROM's store zero at address zero. Moreover, the ROM's within RAC1, RAC2, and RAC3 also store a zero at location 15 (0xF). Let us recall that location 0xF in a distributed arithmetic ROM stores the sum of all elements of the constant vector (Fig. 3). We observe from (1) that all the rows of the 4×4 coefficient matrix except for the top row have elements that sum up to zero. Therefore, locations 0xF within those corresponding three ROM's store a zero. As a result of the observation above, we conclude that equal MSB's among pel sums have no effect on the total outcome of RAC1–RAC3 and can therefore be discarded from the computation. The boxes labelled *MSB equality detection* detect the first different bit among pel sums starting from the most significant bit position. This information is used to produce the qualifying pulse of Fig. 3 and only enable RAC's 1–3 when the address at the corresponding ROM inputs refers to a non-zero-valued ROM location. An example of MSB rejection is shown in Fig. 4(a).

RAC's 1–3 therefore need less than eight cycles on average to compute their corresponding dot products. This reduced duty cycle is exploited with the mechanism outlined in Fig. 3 that disables the RAC's in question in the presence of irrelevant inputs. We note that RAC0 has nonzero content at location 0xF (4A according to (1)) and therefore does not participate in this scheme. It is always clocked.

A similar method (but not 100% equivalent) has been applied to the pel differences $x_0 - x_7, x_1 - x_6, x_2 - x_5, x_3 - x_4$ [(2)] at the bottom half of Fig. 2 (both stages). Equation (2) implies that RAC's 4–7 do not store zeroes at locations 0xF because the rows of the corresponding coefficient matrix do not sum to zero. Yet, another important observation implies that MSB rejection can be applied. Differences of highly correlated samples are very likely to result in small positive or negative values. Obviously, sign extension bits (either 0 for positive or 1 for negative values) do not affect the distributed arithmetic final results. The boxes labelled *Sign Extension Detection* detect the first different non-sign extension bit among the four pel differences. This information is used to reduce the duty cycle of RAC's 4–7 in a fashion equivalent to the method applied to the pel sums. An example of sign extension detection is shown in Fig. 4(b).

The MSB rejection method can reduce substantially the number of arithmetic operations required in the presence of correlated inputs without loss in arithmetic precision. To quantify this statement, we have compared the total number of additions per block required by our MSBR-enhanced DCT algorithm with that of the conventional Chen algorithm [9] (distributed arithmetic implementation) that performs a constant number of operations per block. Fig. 5 displays the number of additions per block required for the first 1000 blocks of test image *Peppers*. The results plotted only include computation savings due to MSB rejection. The graph also displays the number of additions required for the distributed arithmetic version of the Chen algorithm. We observe that our MSBR-enhanced DCT algorithm requires about 40% fewer additions than the standard Chen algorithm. We note that both computation algorithms displayed in Fig. 5 include an equal number of ROM accesses in addition to the required additions. MSBR also results in reduced ROM accesses by the same percentage (40%).

B. Row–Column Classification

As opposed to MSBR, row–column classification (RCC) reduces the overall signal activity by introducing a small error in the arithmetic computation. RCC sets a dynamic upper bound on the number of clocks that each RAC in Fig. 2 will use to compute its corresponding dot product. As described in Section II, the DA operation (MSB first) can be thought of as a successive approximation operation where each cycle an extra bit from each element of the data vector is used to refine the final result. The RAC cycle upper bounds are implemented by

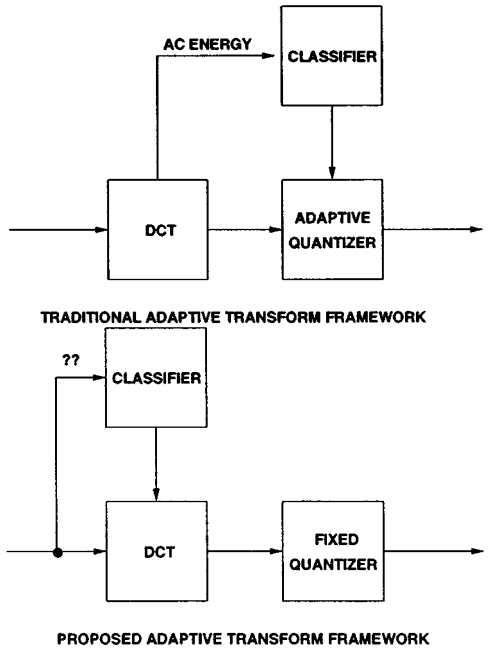

Fig. 6. Traditional versus proposed adaptive transform framework.

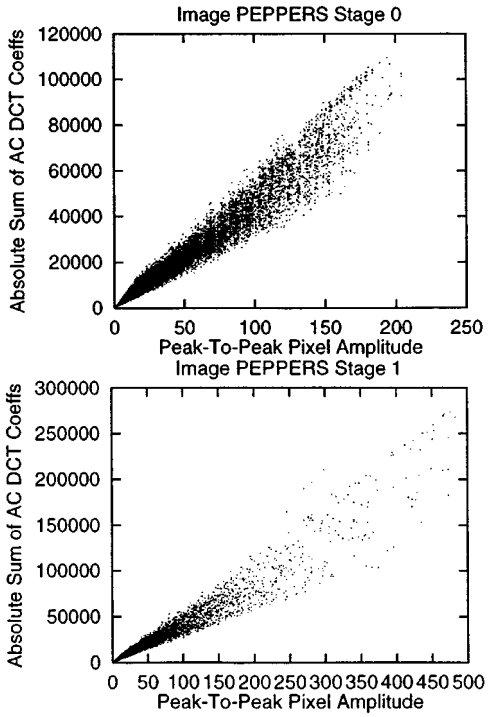

Fig. 7. Correlation between proposed and standard classifier.

user-programmable state registers within the chip and coupling to the preexistent (for MSBR) accumulator clock gating mechanism. The state registers store a clock mask, which is ANDed with the clock mask generated by the MSB rejection mechanism. The cycle upper bounds refer to the maximum number of clock cycles a RAC is being clocked *after* MSB rejection has eliminated common most significant bits from the serial address registers. The user has serial access to the state registers through an IEEE Standard 1149.1-1990 test access port (TAP). When the maximum number of cycles for a particular RAC has been reached and there are still data bits that await processing, the RAC simply powers down and uses its current dot product as an approximation of the final result. Please note that due to our MSB-first implementation of the distributed arithmetic unit (Fig. 3), stopping a RAC before it reaches its final value introduces scaling by a factor 2^{-n}, where n is the number of remaining cycles required to complete the computation using full precision. An additional shift-only feedback path has been provided in each RAC (Fig. 3), which does not engage either the ROM nor the adder in order to undo this scaling. Dynamic adaptivity in this precision reduction mechanism is introduced for image-quality maximization. In order to minimize the truncation noise introduced by the cycle upper bounds described above, we employ a row classifier for the first 1-D stage and a column classifier in the second 1-D stage. The idea is to have increased cycle upper bounds for rows (columns) that exhibit high "activity" (low correlation) and decreased bounds for rows (columns) exhibiting low activity (high correlation). Chip-level simulation results indicated that such a scheme will reduce the mean square error introduced for constant average RAC duty cycle.

Block classifiers proposed in the literature [12]–[14] cannot be applied in the present case because they all assume the availability of ac DCT coefficients and are a function of such coefficients. Fig. 6 illustrates the difference between previously proposed classification schemes and the current approach. The difference stems form the fact that we wish to use classification as a means to reduce computation while keeping quantization parameters constant as opposed to using classification for adapting quantization step sizes.

Our requirements for an appropriate classifier geared to DCT computation reduction are the following:

- The classifier must be a function of the space-domain pels and not a function of the transform-domain coefficients.
- The classifier must be substantially correlated to prior proven classifiers such as the absolute sum of ac coefficients [12].
- The classifier must be a simple and easily computed function of the space-domain pels. It should not add substantially to the total computational load; otherwise energy savings will not be achieved.

We found that the peak-to-peak pixel amplitude per each row (PPA) has the potential to meet all three requirements. It is a function of the space-domain pels $(\max(x_i) - \min(x_i))$ and is easy to compute. Furthermore, it exhibits substantial correlation with a widely used classifier

$$\sum_{i=1}^{N} |X_i| \tag{4}$$

where X_i are the DCT block spectral coefficients and $N = 64$.

Fig. 7 shows a scatter plot for the natural image *Peppers* that indicates substantial correlation between the proposed and standard classifier for both DCT stages. Table I lists the sample correlation between the two classifiers for a set of 11 test images. Fig. 8 shows the cumulative distribution function for the PPA classifier for both DCT stages. The data have been collected

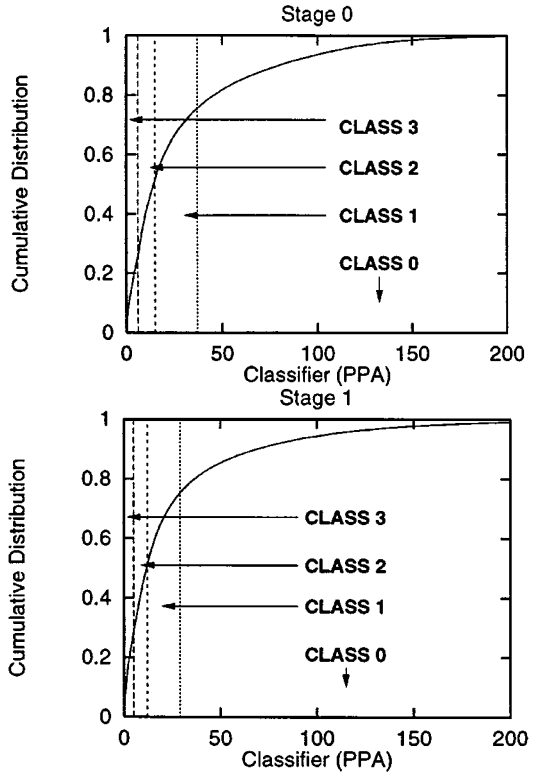

Fig. 8. Image row/column classification threshold determination.

TABLE I
SAMPLE CORRELATION OF PPA VERSUS AC DCT COEFFICIENT SUM CLASSIFIER FOR 11 TEST IMAGES

Image	Stage 0	Stage 1
CATHEDRAL	0.98763	0.98656
COUPLE	0.98429	0.98490
EUROPE	0.98423	0.98438
FLOWER	0.98123	0.97977
GIRL	0.97998	0.98307
LENA	0.98016	0.97932
MANDRILL	0.97808	0.98540
PEPPERS	0.97921	0.97519
PLANE	0.98647	0.98457
VAN GOGH	0.98624	0.98731
WATER	0.98431	0.98930

from the set of 11 images listed in Table I. We choose to divide all rows into four different classes defined by the 25, 50, and 75% ordinates of the distribution in agreement with [13]. Such classification results in equally populated classes. The PPA thresholds for each class are 6, 15, and 37 for stage 0 and 5, 12, and 29 for stage 1. A systematic stepwise optimal procedure has been developed [15] for tabulation of cycle upper bound limits per RAC per class versus desired output quality (PSNR). This procedure indicated that the cycle upper bounds listed in Table II provide high PSNR's for a set of natural images. RAC0 in both stages has no cycle limit whatsoever and always carries its computation to full precision due to its maximum visual significance.

TABLE II
RAC CYCLE UPPER LIMITS (MAXIMUM COMPUTATION BITWIDTH) PER CLASS

	RAC1	RAC2	RAC3	RAC4	RAC5	RAC6	RAC7
Class 0	8	6	6	4	4	3	2
Class 1	8	6	6	4	4	0	0
Class 2	6	4	4	0	0	0	0
Class 3	4	0	0	0	0	0	0

RCC is a simple and elegant way of trading off power dissipation and image quality with minimal circuit overhead (monitoring and clock gating mechanisms) and minimal image quality degradation in the mean-square-error sense. The PPA classifier is computed at each stage on a row (stage 0) or column (stage 1) basis using the circuit of Fig. 9. This subsystem is contained within the row/column classifier blocks of Fig. 2. The two comparators ensure that register D0 holds the row (column) maximum element and D1 holds the row (column) minimum element. The subtractor computes the maximum absolute difference (PPA classifier). The classifier is compared downstream with three user-supplied thresholds, and the row (column) class identifier (0–3) is computed. The class identifier is used to select the appropriate set of RAC cycle upper limits (Table II) for the distributed arithmetic computations.

IV. CIRCUIT DESIGN

The DCT chip uses a total of 16 20-bit adders, one in each RAC unit. The 20-bit carry-bypass adder shown in Fig. 10 is used. The bypass adder has been chosen because it has a short critical path at the expense of only a small increase in area. Its operation is very simple. The full adders are divided among cascaded groups of four. Although in a carry-bypass adder minimum delay is obtained when the adder groups have variable numbers of full adders per block, we decided not to implement this scheme mainly for layout considerations. When all the propagate signals within a certain full adder group are high, then the carry from the previous group is forwarded instead of the locally generated one without any change in I/O behavior. The critical path of the adder in Fig. 10 is exercised when a carry is generated within the (A0, B0, S0) full adder and propagated all the way to the (A19, B19, S19) full adder. This critical path is approximately eight full adders plus four multiplexer delays. The addition of four more extra bits of arithmetic bitwidth adds a single multiplexer delay to the critical path.

The full adder used is shown in Fig. 11. It has been selected because the propagate (P) signal necessary for the carry-bypass operation is computed explicitly, and due to its very small carry-in to carry-out delay. While cascading the full adders to form the 20-bit adder of Fig. 10, we exploit the fact that a full adder is fully complementary (input complements produce output complements) to avoid unnecessary inversions in the carry propagation path.

Fig. 12 shows the circuit that detects common most significant bits and computes a clock mask that is used to disable RAC

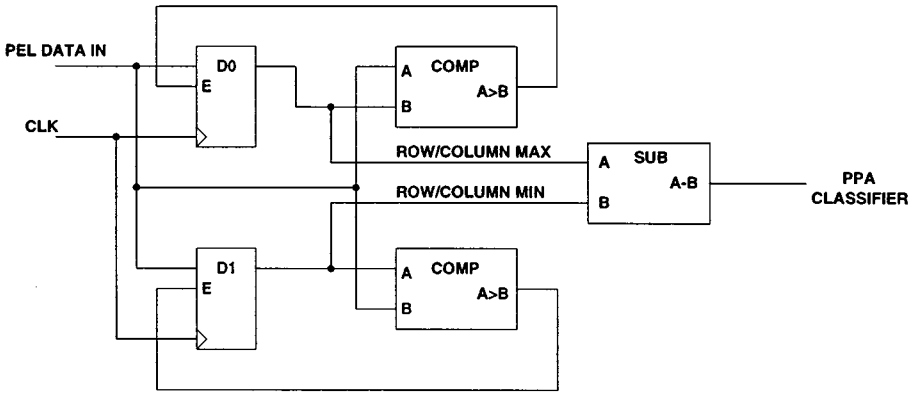

Fig. 9. Row/column classifier implementation.

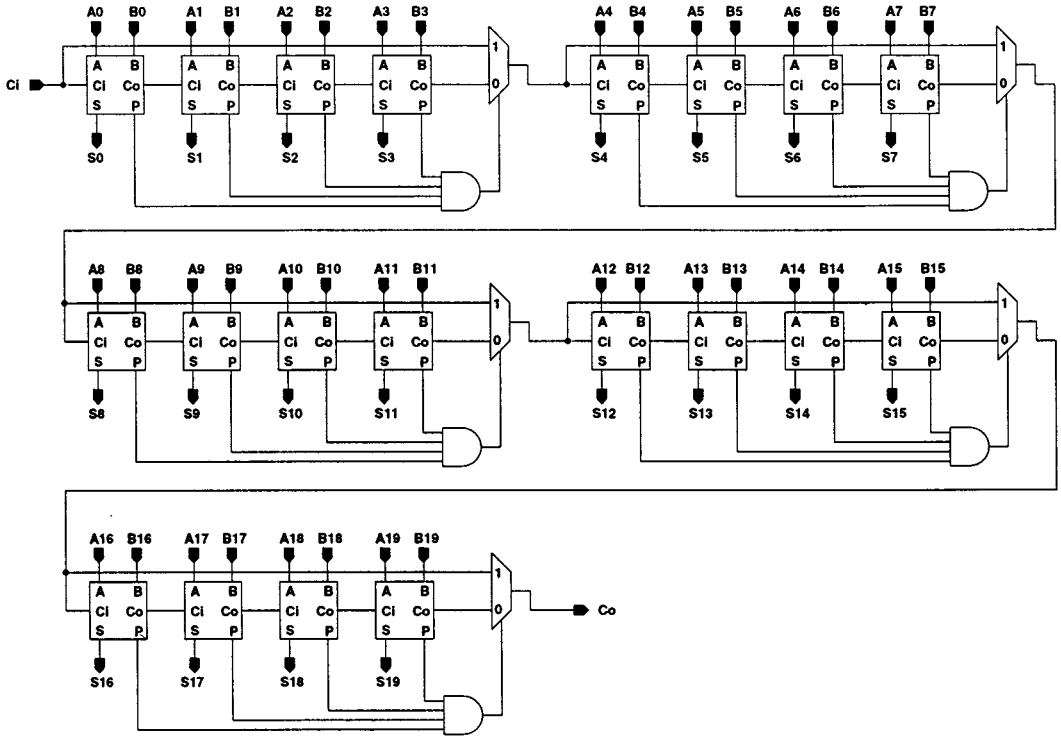

Fig. 10. 20-bit carry-bypass adder.

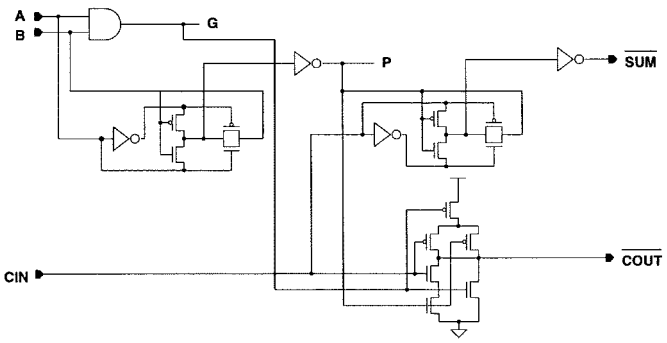

Fig. 11. Full adder used in the DCT chip.

units 1, 2, and 3 of each stage when the ROM address inputs do not affect the computation. This circuit operates on the four sums (A7-0, B7-0, C7-0, D7-0) of the "butterfly" additions of each stage (Fig. 2). Transistors N1, N2, N3, N4 and N5, N6, N7, N8 detect the presence of all ones or all zeroes at the input, respectively. The cascading of the outputs through the NAND gates and inverters activates all the remaining downstream mask bits as soon as the first bit position is found where 8-bit-wide inputs A, B, C, and D have different bits. The computed clock mask (CLKMASK) bits indicate when RAC units 1–3 should be clocked (1 in the corresponding bit position) and when not (0 in the corresponding bit position.)

V. Physical Design and Test Results

The chip has been fabricated in a triple-metal 0.6-μm 5-V CMOS process. The annotated chip microphotograph is shown in Fig. 13. The blocks labelled "MSB rejection" and "sign detection," in addition to the clock mask generation mechanisms, include distributed control circuits and clock generators and buffers for the RAC units of both stages. Moreover, each contains a 128-bit JTAG register (visible as a dark rectangular

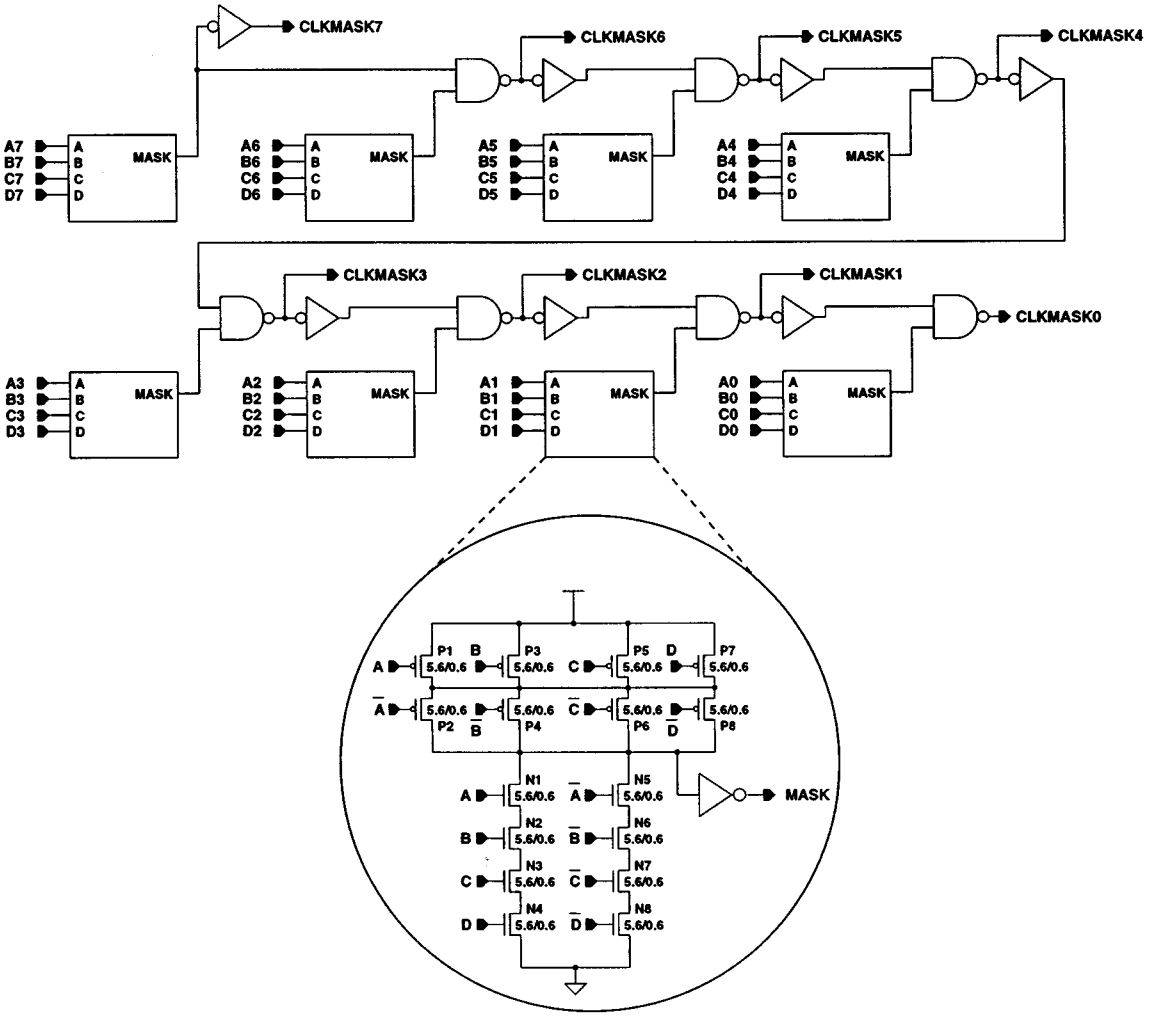

Fig. 12. MSB rejection circuit.

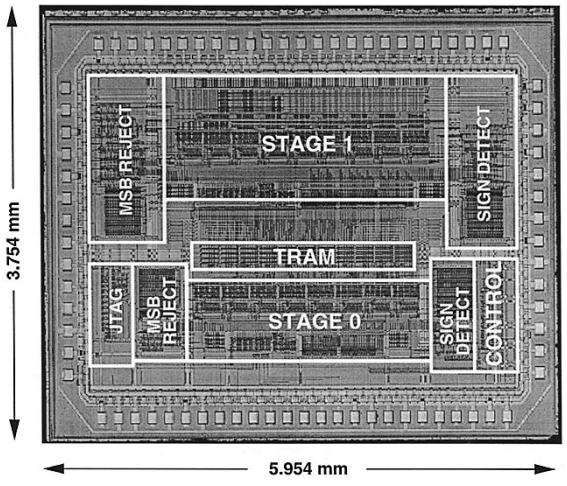

Fig. 13. DCT chip microphotograph.

box within each block) that stores cycle upper bounds (four per class per RAC). These registers are all accessible through the IEEE 1149.1-1990 TAP. The block labelled "JTAG" contains the IEEE 1149.1 instruction register and the finite-state machine (TAP controller), which controls serial access to all JTAG-accessible registers. The core area of the DCT chip (not including the I/O pad frame) measures 14.5 mm^2 and includes approximately 120-K transistors. The chip specifications are summarized in Table III. The DCT chip is functional over a wide range of frequencies and power supplies, as can be seen from the schmoo plot of Fig. 14.

More than 150 000 separate power measurements on a block-by-block basis have been taken using natural images in addition to artificially generated data in order to fully characterize the chip power dissipation. The results of our power measurements (average ± 1 standard deviation) on a 64-element block basis are plotted in Fig. 15 versus the sample standard deviation of the 64 block elements. The image block frequency histogram versus standard deviation is also shown on the same plot. We observe that according to the design goal, power dissipation shows strong dependence with data correlation (more MSB rejection and less arithmetic activity).

Fig. 16 compares the average power of the DCT chip for different types of stimuli ranging from fully correlated (constant) blocks to random blocks with maximum variance. The "IMAGE BLOCKS" bars are power averages when the chip is stimulated from 11 natural test images. We observe that MSB rejection itself can be responsible for up to 55% power savings, with 22%

TABLE III
PROCESS AND DCT CHIP SPECIFICATIONS

Process	0.6μm CMOS (0.6μm drawn), 3ML 5V
VTN	0.75V
VTP	-0.82V
TOX	14.8 nm
Supply	1.1-3 V
Frequency	2-43 MHz
Sample Rate	2-43 Msamples/sec
Power	4.38 mWatts @ 1.56 V, 14 MHz
Area	14.5 mm^2
Transistors	120K

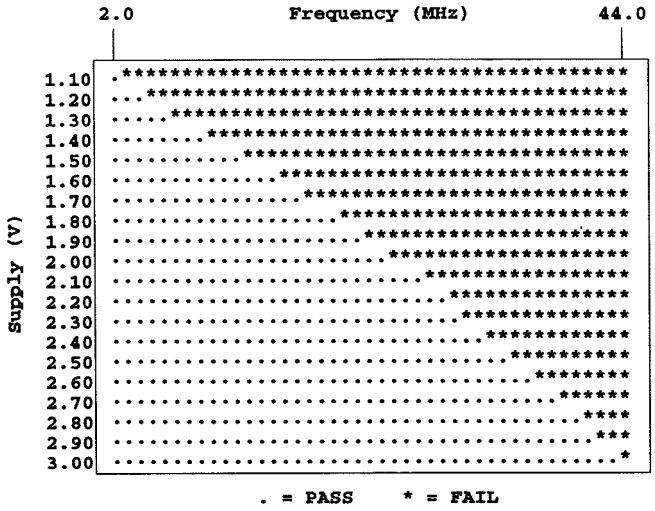

Fig. 14. DCT chip schmoo plot.

being more typical for image data. The savings figure will be increased for differential video data (i.e., MPEG) due to increased correlation (more zeroes and small magnitude numbers present.)

Finally, we wish to establish the relationship between power dissipation and image coding quality. We used the *Peppers* image and picked 11 sets of cycle limits than span the entire achievable PSNR range for this image (23.99–44.84 dB). We measured the power that image *Peppers* dissipates for each one of the 11 limit sets and plotted the results versus PSNR in Fig. 17.

Fig. 18 displays the actual compressed images for three (power, PSNR) datapoints of Fig. 17. Figs. 17 and 18 establish our claim that the present chip trades off image quality and power dissipation.

We have used an internally developed power estimator [16], [11] to calculate the chip power dissipation on a block basis. Fig. 19 shows the power dissipated by the DCT chip partitioned among the top-level design modules. The chip has been stimulated with the first 1000 blocks of test image *Peppers* using the cycle bounds of Table II. The simulation includes estimated interconnect capacitance. The total power estimated for this particular data set is 7.4 mW at 1.56 V and 14 MHz.

We observe that according to the power estimation results, the main clock buffer dissipates most of the chip power (almost 37%). Almost 60% of the total clock buffer power is due to wiring parasitics. The chip pure computation power (stages 0 and 1) are estimated at about 42% of the total (Fig. 19). We observe that the cost of the MSB rejection control logic is approximately 3-4% of the total (about half of the MSBR+Inhi-

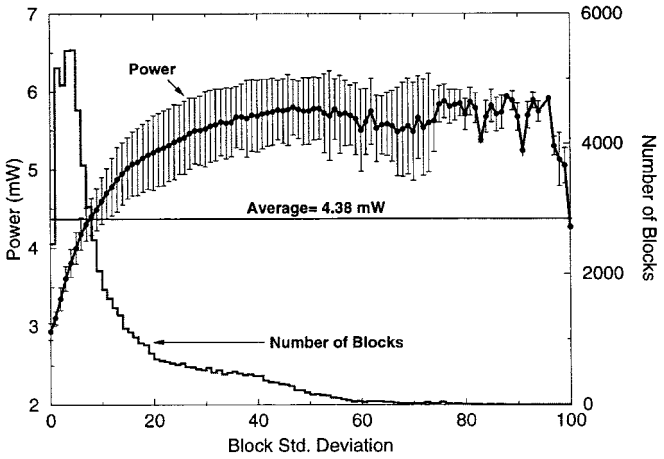

Fig. 15. DCT chip power versus pel standard deviation. Chip measured power at 1.56 V, 14 MHz. The cycle limits used are shown in Table II. This plot includes all blocks of 11 natural test images.

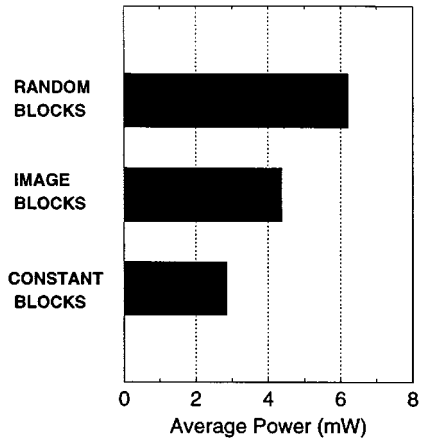

Fig. 16. DCT chip average power comparison for different stimuli. Chip measured power at 1.56 V, 14 MHz.

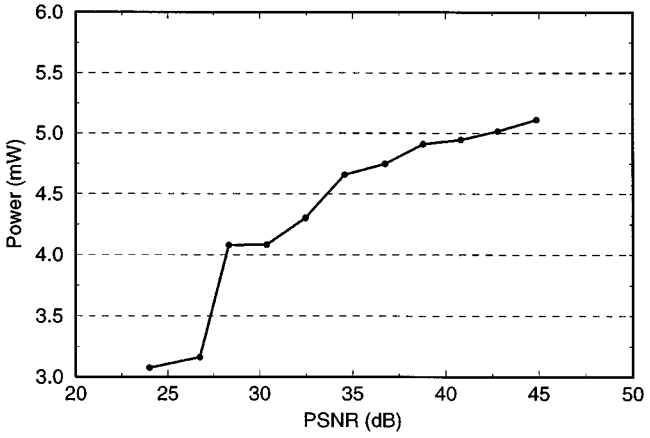

Fig. 17. DCT chip average power versus compressed image quality. Chip measured power at 1.56 V, 14 MHz. The power measurements are for test image *Peppers*.

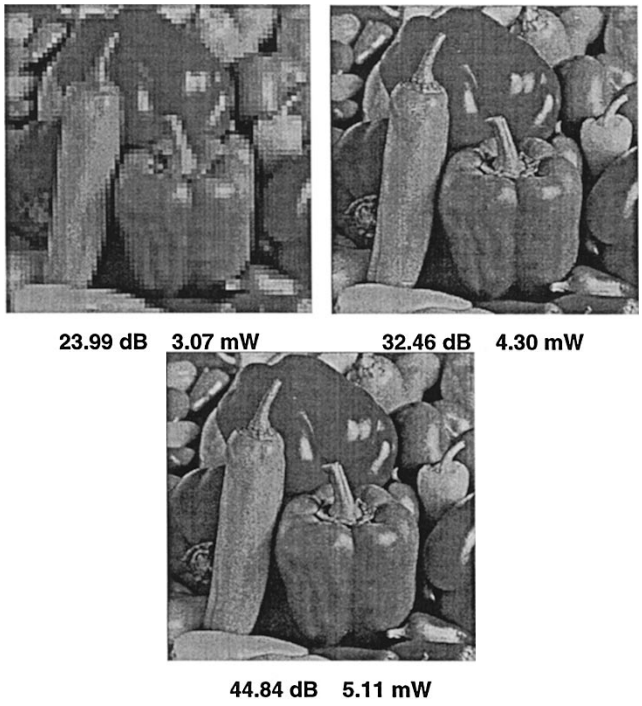

Fig. 18. Compressed image quality and power. The displayed images constitute three of the data points of Fig. 17.

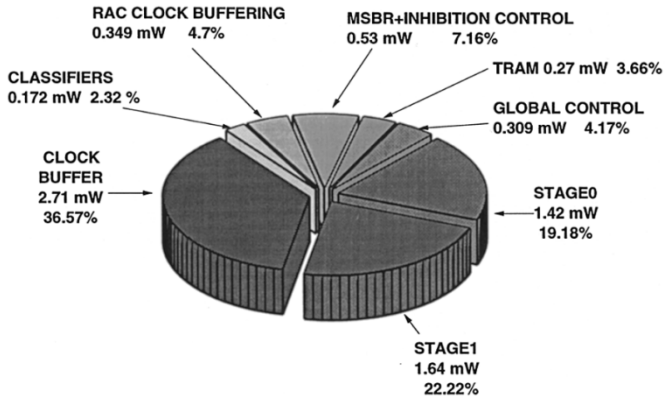

Fig. 19. DCT chip power estimation results.

TABLE IV
ENERGY EFFICIENCY COMPARISON AMONG DCT/IDCT CHIPS

Chip	Sw-Cap/sample
Matsui et al. [17]	375 pF
Bhattacharya et al. [19]	479 pF
Kuroda et al. [18] (scaled to same feature size)	417 pF
Present DCT chip	128 pF

bition control sector), whereas the computation inhibition plus row/column classification is at 5–6% of the total (about half of the MSBR+Inhibition control sector plus the row/column classifiers.)

The DCT chip is more energy efficient than similar chips that have appeared in the literature from a switched capacitance perspective, as Table IV suggests. We observe that the present chip exhibits less switched capacitance by a factor of three when compared to past DCT processors adjusted for similar process feature sizes. The activity reduction methods account for a significant portion of this reduction. The remaining savings are attributed to optimized internal arithmetic bitwidths, additional clock gating, and reduced on-chip latency. Chips [17] and [18] impose 112 cycles of latency from chip input to chip output. The latency of the present DCT chip is 97 cycles.

VI. CONCLUSION

The VLSI implementation of the DCT chip has demonstrated the suitability of distributed arithmetic structures for low power. Such structures can reject computation-irrelevant bits in a very elegant fashion using minimal control overhead and at a very fine granularity. Moreover, the ROM and accumulator arithmetic units can be used in an approximate processing framework with minimum extra hardware control overhead. Simply stopping the clock of an RAC before it completes its dot product computation produces an approximation of its final result with almost linear power savings. The DCT chip has demonstrated this property by producing approximate results for nonvisually significant spectral coefficients and for low activity input data.

In our VLSI implementation, we have observed that power savings due to MSBR can be as high as 55% (random blocks versus fully correlated blocks), with 22% being more typical for still images. The additional power cost of MSBR is 3–4% of the total chip power. Differential video will result in higher savings due to the increased presence of zero-valued data. RCC adds an additional 15% of power savings for minimal PSNR degradation at the expense of 5-6% more power in additional control logic. Much higher power savings can be achieved if we are willing to tolerate more image degradation. Both MSBR and and RCC account for about 40% of power savings for still images at an additional combined overhead of 10% for a net power reduction of 30%. We expect higher power savings for differential video coding.

REFERENCES

[1] L. Nielsen and J. Sparso, "A low-power asynchronous data-path for an FIR filter bank," in *Proc. 2nd Int. Symp. Advanced Research in Asynchronous Circuits and Systems*, Mar. 1996, pp. 197–207.
[2] ——, "An 85 μW asynchronous filter bank for a digital hearing aid," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 1998, pp. 108–109.
[3] J. Ludwig, S. Nawab, and A. Chandrakasan, "Low-power digital filtering using approximate processing," *IEEE J. Solid-State Circuits*, vol. 31, pp. 395–400, Mar. 1996.
[4] P. Larsson and C. Nikol, "Self-adjusting bit precision for low-power digital filters," in *Proc. Symp. VLSI Circuits*, June 1997, pp. 123–124.
[5] C. Nikol, P. Larsson, K. Azadet, and N. O'Neill, "A low-power 128-tap digital adaptive equalizer for broadband modems," *IEEE J. Solid-State Circuits*, vol. 32, pp. 1777–1789, Nov. 1997.
[6] A. Peled and B. Liu, "A new hardware realization of digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-22, pp. 456–462, Dec. 1974.
[7] S. A. White, "Applications of distributed arithmetic to digital signal processing: A tutorial review," *IEEE Acoust., Speech, Signal Processing Mag.*, pp. 4–19, July 1989.
[8] R. Amirtharajah, T. Xanthopoulos, and A. Chandrakasan, "Power scalable processing using distributed arithmetic," in *1999 Symp. Low Power Electronics and Design Dig. Tech. Papers*, Aug. 1998, pp. 170–175.
[9] W. H. Chen, C. H. Smith, and S. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Trans. Commun.*, vol. COM-25, pp. 1004–1009, Sept. 1977.
[10] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. New York: Academic, 1990.

[11] T. Xanthopoulos and A. Chandrakasan, "A low-power IDCT macrocell for MPEG2 MP@ML exploiting data distribution properties for minimal activity," *IEEE J. Solid-State Circuits*, vol. 34, pp. 693–703, May 1999.
[12] J. Gimlett, "Use of "activity" classes in adaptive transform image coding," *IEEE Trans. Commun.*, vol. COM-23, pp. 785–786, July 1975.
[13] W. H. Chen and H. Smith, "Adaptive coding of monochrome and color images," *IEEE Trans. Commun.*, vol. COM-25, pp. 1285–1292, Nov. 1977.
[14] R. Mester and U. Franke, "Spectral entropy-activity classification in adaptive transform coding," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 913–917, June 1992.
[15] T. Xanthopoulos, "Low Power Data-Dependent Transform Video and Still Image Coding," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, 1999.
[16] T. Xanthopoulos, Y. Yaoi, and A. Chandrakasan, "Architectural exploration using Verilog-based power estimation: A case study of the IDCT," in *Proc. 34th Design Automation Conf. (DAC'97)*, June 1997, pp. 415–420.
[17] M. Matsui, H. Hara, K. Seta, Y. Uetani, L. Kim, T. Nagamatsu, T. Shimazawa, S. Mita, G. Otomo, T. Oto, Y. Watanabe, F. Sano, A. Chiba, K. Matsuda, and T. Sakurai, "200 MHz video compression macrocells using low-swing differential logic," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 1994, pp. 76–77.
[18] T. Kuroda, T. Fujita, T. Nagamatu, S. Yoshioka, F. Sano, M. Norishima, M. Murota, M. Kako, M. Kinugawa, M. Kakumu, and T. Sakurai, "A 0.9 V, 150-MHz, 10-mW, 4 mm^2, 2-D discrete cosine transform core processor with variable-threshold-voltage (VT) scheme," *IEEE J. Solid-State Circuits*, vol. 31, pp. 1770–1777, Nov. 1996.
[19] A. K. Bhattacharya and S. S. Haider, "A VLSI implementation of the inverse discrete cosine transform," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 9, no. 2, pp. 303–314, 1995.

Thucydides Xanthopoulos (S'91) received the S.B., S.M., and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 1992, 1995, and 1999, respectively.

He is currently an Independent Technical Consultant in the areas of high-speed clocking and deep submicrometer IC design. He is a member of the Technical Program Committee of the 2000 International Symposium on Low Power Electronics and Design. His research interests include low-power VLSI design, embedded DSP architectures, CAD for low power, and clocking for high-speed systems.

Dr. Xanthopoulos is a member of the Association of Computing Machinery, Tau Beta Pi, and Eta Kappa Nu. In 1996, he received a National Semiconductor Graduate Fellowship.

Anantha P. Chandrakasan (S'87–M'95) received the B.S, M.S., and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1989, 1990, and 1994, respectively.

Since September 1994, he has been with the Massachusetts Institute of Technology, Cambridge, and is currently an Associate Professor of electrical engineering and computer science. He held the Analog Devices Career Development Chair from 1994 to 1997. His research interests include the ultra-low-power implementation of custom and programmable digital signal processors, distributed wireless sensors, multimedia devices, emerging technologies, and CAD tools for VLSI. He is a coauthor of *Low Power Digital CMOS Design* (Norwell, MA: Kluwer Academic, 1995) and a coeditor of *Low Power CMOS Design* (New York: IEEE Press, 1998). He has served on the technical program committee of various conferences, including ISSCC, VLSI Circuits Symposium, DAC, and ISLPED. He has served as a technical program Cochair for the 1997 International Symposium on Low-Power Electronics and Design (ISLPED), VLSI Design'98, and the 1998 IEEE Workshop on Signal Processing Systems; and as a General Cochair of the 1998 ISLPED. He is the Signal Processing Subcommittee Chair for ISSCC'00.

Prof. Chandrakasan is an Associate Editor of the IEEE JOURNAL OF SOLID-STATE CIRCUITS. He received the NSF Career Development Award in 1995, the IBM Faculty Development Award in 1995, and the National Semiconductor Faculty Development Award in 1996 and 1997. He has received several best paper awards, including the 1993 IEEE Communications Society's Best Tutorial Paper Award, the IEEE Electron Devices Society's 1997 Paul Rappaport Award for the Best Paper in an EDS publication during 1997, and the 1999 Design Automation Conference Design Contest Award. He is a member of the Design and Implementation of Signal Processing Systems Technical Committee of the Signal Processing Society.