

# Parallel-Pipeline $8 \times 8$ Forward 2-D ICT Processor Chip for Image Coding

Gustavo A. Ruiz, Juan A. Michell, and Angel M. Burón

**Abstract**—The Integer Cosine Transform (ICT) presents a performance close to Discrete Cosine Transform (DCT) with a reduced computational complexity. The ICT kernel is integer-based, so computation only requires adding and shifting operations. This paper presents a parallel-pipelined architecture of an  $8 \times 8$  forward two-dimensional (2-D) ICT(10, 9, 6, 2, 3, 1) processor for image encoding. A fully pipelined row-column decomposition method based on two one-dimensional (1-D) ICTs and a transpose buffer based on D-type flip-flops is used. The main characteristics of 1-D ICT architecture are high throughput, parallel processing, reduced internal storage, and 100% efficiency in computational elements. The arithmetic units are distributed and are made up of adders/subtractors operating at half the frequency of the input data rate. In this transform, the truncation and rounding errors are only introduced at the final normalization stage. The normalization coefficient word length of 18-bit (13-bit effective) has been established using the requirements of IEEE standard 1180–1990 as a reference. The processor has been implemented using standard cell design methodology in 0.35- $\mu\text{m}$  CMOS technology, measures 9.3 mm<sup>2</sup>, and contains 12.4 k gates. The maximum frequency is 300 MHz with a latency of 214 cycles (260 cycles with normalization).

**Index Terms**—Integer cosine transform, image compression, multiplication free DCT, parallel pipelined architectures, VLSI.

## I. INTRODUCTION

THE Discrete Cosine Transform (DCT) is widely considered to provide the best performance for transform coding and image compression [1]. The DCT, which is known to be the closest to the ideal energy compaction transform, has become an international standard for sequential codecs as JPEG, MPEG, H.261, H.263, etc. [2], [3]. However, DCT matrix elements contain real numbers represented by a finite number of bits, which inevitably introduce truncation and rounding errors during computation. Thus, many applications which use this transform can be classified under the heading of “lossy” encoding schemes. This implies that the reconstructed image is always only an approximation of the original image.

VLSI implementation of DCT using floating-point arithmetic is highly complex and requires multiplications [4]–[15]. Different multiplication-free algorithms, which are approximations to the DCT, have been proposed in order to reduce implementation complexity [16]–[22]. In some cases, they can be used

for lossless compression applications since the round-off error can be completely eliminated. In these algorithms, coefficients are scaled or approximated [16], [17] so that the floating-point multiplication can be implemented efficiently by binary shifts and additions. The binDCT algorithm proposed in [18] introduces the integer lifting scheme to derive coefficients of the form  $k/2^m$ . The integer DCT (intDCT) [20] is based on the Walsh–Hadamard transform and the integer lifting scheme and presents a more systematic way than binDCT to lift the transform. The Integer Cosine Transform (ICT) [21], [22] is generated applying the concept of dyadic symmetry and presents a similar performance and compatibility with the DCT [21], [23], [24] using either a programmable processor or dedicated hardware. The ICT basis components are integers so they do not require floating-point multiplications, as these are substituted by fixed-point addition and shifting operations, as they have more efficient hardware implementation.

One of the first practical applications of ICT was the image compression scheme of the NASA/JPL Galileo spacecraft [25]. In this case, the lower complexity of the ICT was necessary due to the limited performance of 1970s microprocessor technology. Subsequently, it has been used in other applications as image transforms [28] and video compression [30]. An analysis carried out in real images on various ICT kernels showed that ICT(10, 9, 6, 2, 3, 1) is the one with the best performance in terms of efficiency, complexity, and mean square-error [21]. Two very different VLSI architectures have been implemented based on this transform: a two-dimensional (2-D) ICT hardware realization with three chips on a gate array technology at a clock rate of 3 MHz [28] and an asynchronous 1-D ICT [29] with a data input rate of 50 MHz. In both implementations, a single central arithmetic unit is used, based on a multiplier made up of multiplexers and adders, and an adder/subtractor.

This paper describes the architecture of an  $8 \times 8$  forward 2-D ICT processor chip for image coding. This processor has been conceived using a standard cell design methodology and manufactured with a 0.35- $\mu\text{m}$  CMOS CSD 3M/2P 3.3-V process [33]. The circuit includes 12 446 cells (6757 of which are flip-flops) on a 9.3-mm<sup>2</sup> die; more details about chip implementation can be found in [31] and [32]. The 2-D ICT is calculated using the separability property so that its architecture is made up of two 1-D ICT(10, 9, 6, 2, 3, 1) processors and a transpose buffer as intermediate memory. This transpose buffer presents a regular structure based on D-type flip-flops with a serial input/output data-flow very adequate for pipeline architectures. The processor uses a low latency data flow that minimizes the internal memory and a parallel-pipelined architecture to attain an operational frequency of 300 MHz.

Manuscript received July 8, 2003; revised February 4, 2004. This work was supported by the Spanish Ministry of Science and Technology under Grant TIC2000-1289. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chaitali Chakrabarti.

The authors are with the Departamento de Electrónica y Computadores, Facultad de Ciencias, Universidad de Cantabria, 39005 Santander, Spain (e-mail: ruizrg@unican.es).

Digital Object Identifier 10.1109/TSP.2004.840682

Moreover, the arithmetic units are based on 100% efficient adders/subtractors operating at half the frequency of the data input rate. The output coefficients can be selected with or without normalization. In the latter case, the normalization coefficients' word length must be 18 bit, of which only 13 bits are necessary, if the specifications of the IEEE standard 1180–1990 [35] are adhered to.

The paper is organized as follows: A decomposition of the ICT(10, 9, 6, 2, 3, 1) to obtain a signal flow chart, which leads to an efficient hardware implementation, is presented in Section II. In Section III, a pipeline structure is proposed of the 1-D eight-order J(10, 9, 6, 2, 3, 1) transform, based on three processors operating in parallel with adders/subtractors combined with wired-shift operations as the only arithmetic elements. Finally, the 8 × 8 forward 2-D ICT processor architecture, and comparisons with other previous ICT/DCT processors are described in Section IV.

## II. DECOMPOSITION OF THE ICT

The ICT was derived from DCT by the concept of dyadic symmetry [21]. Let  $\mathbf{T}$  be the matrix that represents the order- $N$  DCT. The  $m$ th element of this matrix is defined as

$$\mathbf{T}_{nm} = \left(\frac{1}{N}\right)^{1/2} \left[ k_m \cos\left(\frac{m(n+1/2)\pi}{N}\right) \right]_{m, n = 0, 1, \dots, N-1} \quad (1)$$

where

$$k_m = \begin{cases} 1, & \text{if } m \neq 0 \text{ or } N \\ \frac{1}{\sqrt{2}}, & \text{if } m = 0 \text{ or } N. \end{cases} \quad (2)$$

The order-8 DCT kernel derived from (1) can be expressed in matrix notation as

$$\mathbf{T} = \mathbf{K}\mathbf{J} \quad (3)$$

where  $\mathbf{K}$  is the normalization diagonal matrix, and  $\mathbf{J}$  an orthogonal matrix made up of the basis components of DCT.  $\mathbf{J}$  is defined as

$$\mathbf{J} = \begin{bmatrix} g & g & g & g & g & g & g & g \\ a & b & c & d & -d & -c & -b & -a \\ e & f & -f & -e & -e & -f & f & e \\ b & -d & -a & -c & c & a & d & -b \\ g & -g & -g & g & g & -g & -g & g \\ c & -a & d & b & -b & -d & a & -c \\ f & -e & e & -f & -f & e & -e & f \\ d & -c & b & -a & a & -b & c & -d \end{bmatrix} \quad (4)$$

$a, b, c, d, e, f,$  and  $g$  being constants;  $g$  is 1. The dyadic symmetry present in  $\mathbf{J}$  reveals that to ensure their orthogonality, the constants  $a, b, c,$  and  $d$  must satisfy the following only condition

$$ab = ac + bd + cd. \quad (5)$$

Furthermore, if  $\mathbf{J}$  must be similar to DCT, it implies that

$$a \geq b \geq c \geq d, \quad \text{and} \quad e \geq f \quad (6)$$

Those  $\mathbf{T}$  matrices that satisfy (5) and (6) also have  $a, b, c, d, e,$  and  $f$  constants, which are integers, are called integer cosine transform and are denoted as ICT( $a, b, c, d, e, f$ ).

The 1-D ICT for a real input sequence  $x(n)$  is defined as

$$\mathbf{X} = \mathbf{T}\mathbf{x} = \mathbf{K}\mathbf{J}\mathbf{x} = \mathbf{K}\mathbf{Y} \quad (7)$$

where  $\mathbf{X}$  and  $\mathbf{x}$  are dimension-8 column matrices, and  $\mathbf{K}$  is the diagonal normalization matrix. Reordering the input sequence and the transform coefficients according to the rules

$$\begin{cases} x'(n) = x(n) & n \in [0, 3] \\ x'(7-n) = x(n+4) & n \in [0, 3] \end{cases} \quad (8)$$

$$\begin{cases} X'(m) = X(\text{Br8}[m]) \\ X'(m+4) = X(2m+1) \end{cases} \quad m \in [0, 3] \quad (9)$$

where  $\text{Br8}[m]$  represents bit-reverse operation of length 8, then the 1-D ICT can be expressed as

$$\mathbf{X}' = \mathbf{T}_R \mathbf{x}' = \mathbf{K}_R \mathbf{J}_R \mathbf{x}' = \mathbf{K}_R \mathbf{Y}'. \quad (10)$$

The reordered basis components of ICT can be expressed as

$$\mathbf{J}_R = \begin{bmatrix} \mathbf{J}_{4e} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{4o} \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \quad (11)$$

$\mathbf{I}_4$  being the dimension-4 identity matrix, and

$$\mathbf{J}_{4e} = \begin{bmatrix} g & g & g & g \\ g & -g & -g & g \\ e & f & -f & -e \\ f & -e & e & -f \end{bmatrix} \quad \text{and} \quad \mathbf{J}_{4o} = \begin{bmatrix} a & b & c & d \\ b & -d & -a & -c \\ c & -a & d & b \\ d & -c & b & -a \end{bmatrix}. \quad (12)$$

Applying the decomposition rules defined in (8) and (9) to the  $\mathbf{J}_{4e}$  matrix results in

$$\mathbf{J}_{4e} = \begin{bmatrix} \mathbf{J}_{2e} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{2o} \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 \\ \mathbf{I}_2 & -\mathbf{I}_2 \end{bmatrix} \mathbf{R}_4 \quad (13)$$

where  $\mathbf{R}_4$  is the reordering matrix of length 4,  $\mathbf{I}_2$  is the dimension-2 identity matrix, and

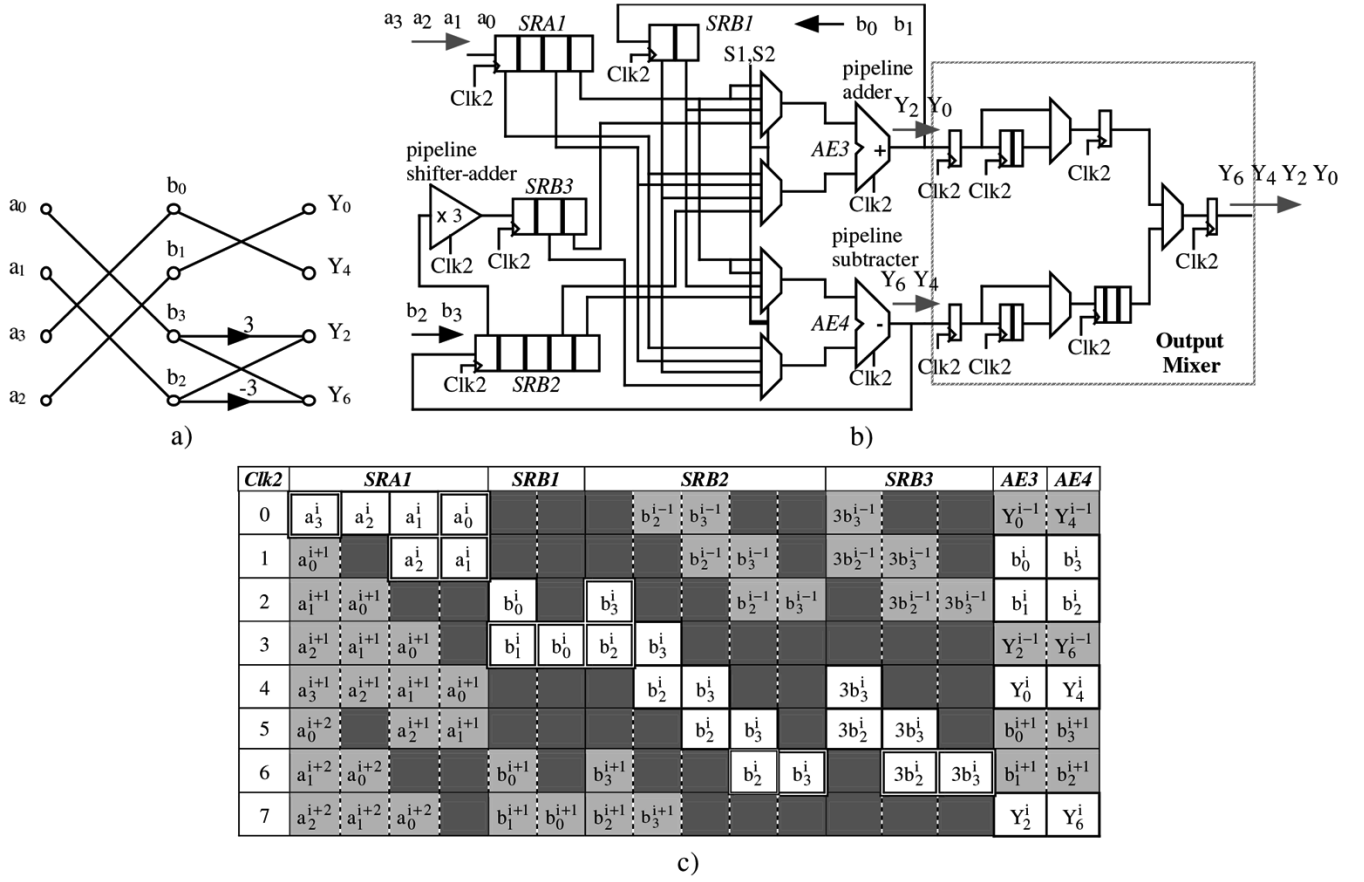
$$\mathbf{J}_{2e} = \begin{bmatrix} g & g \\ g & -g \end{bmatrix} \quad \text{and} \quad \mathbf{J}_{2o} = \begin{bmatrix} e & f \\ f & -e \end{bmatrix}. \quad (14)$$

Fig. 1 shows the signal flow graph obtained by applying the decomposition process to ICT(10, 9, 6, 2, 3, 1). In this case, we have (15), shown at the bottom of the next page.

The  $\mathbf{J}_{4e}$  and  $\mathbf{J}_{4o}$  matrices are defined as

$$\mathbf{J}_{4e} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 3 & 1 & -1 & -3 \\ 1 & -3 & 3 & -1 \end{bmatrix} \quad (16)$$




 Fig. 4.  $J_{4e}$  processor. (a) Signal flow graph. (b) Architecture. (c) Timing diagram.

data of the first computing stage ( $a_0$  to  $a_7$ ) and two parallel processors computing the transformations  $J_{4e}$  and  $J_{4o}$ . These three processors have parallel architecture, allowing the operation frequency to be reduced to  $f_s/2$ , where  $f_s$  is the input data sampling frequency. The final output mixer arranges, in natural form, the coefficient sequence of the ICT at a frequency of  $f_s$ . The control of the processor is very simple and is carried out using four signals: Clk1, external clock at frequency  $f_s$ ; Clk2, internal clock at frequency  $f_s/2$ ; and the multiplexer selection signals S1 at frequency  $f_s/4$  and S2 at frequency  $f_s/8$ . The arithmetic multiplications have been separated into  $\times 2$ ,  $\times 3$  and  $\times 8$  terms so that the arithmetic units are reduced to adders and to subtractors combined with wired-shift operations. All adders/subtractors in the processor are based on the binary carry lookahead (BCL) adder [34] with one stage of pipelined registers in the middle operating at  $f_s/2$  to attain a high-speed addition.

#### A. Input Processor

The input processor shown in Fig. 3 performs the first computing stage with the input sequence data introduced in natural form at frequency  $f_s$ . This input data is stored in a shift register of length 11, and the two multiplexers 4:1 select the data to be processed by AE1 and AE2 in parallel at a subsampling frequency of  $f_s/2$ : AE1 generates the intermediate data  $\langle a_3, a_2, a_1, a_0 \rangle$  and AE2 the  $\langle a_4, a_5, a_6, a_7 \rangle$ .

#### B. $J_{4e}$ Processor

The processor  $J_{4e}$  has been designed to calculate the even coefficients of the 1-D  $J$  transform. From the decomposition procedure established in (13) and (14) applied to (16), we get

$$\begin{bmatrix} Y_0 \\ Y_4 \\ Y_2 \\ Y_6 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 1 & -3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_3 \\ a_2 \end{bmatrix} \\ = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 1 & -3 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_3 \\ b_2 \end{bmatrix} \quad (18)$$

$\langle b_0, b_1, b_2, b_3 \rangle$  being the intermediate data. Operating on (18), we get

$$\begin{bmatrix} Y_0 \\ Y_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} Y_2 \\ Y_6 \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 1 & -3 \end{bmatrix} \begin{bmatrix} b_3 \\ b_2 \end{bmatrix}. \quad (19)$$

Fig. 4(a) shows the signal flow graph obtained from (18) and (19), and Fig. 4(b) shows the proposed architecture. This processor has four shift registers, four multiplexers 4:1, three arithmetic units, and an output mixing to be ordered  $\langle Y_0, Y_2, Y_4, Y_6 \rangle$ . Fig. 4(c) contains the timing diagram specified in number of cycles of Clk2. For the sake of clarity, only the valid data contained in the shift registers and output data of AE3 and AE4 are shown in this diagram. The white cells show

the valid data corresponding to the  $i$ th transform, and the white cells with a two-lined box indicate the shift register data processed by the arithmetic units. The light-gray cells contain data of previous or posterior transforms, and the empty dark-gray cells indicate nonvalid data. The process begins storing the input data  $\langle a_3^i, a_2^i, a_1^i, a_0^i \rangle$  in *SRA1*. *AE3* and *AE4* generate the intermediate data  $\langle b_3^i, b_2^i, b_1^i, b_0^i \rangle$ ;  $b_0^i$ , and  $b_1^i$  that are stored in register *SRB1*, whereas  $b_2^i$  and  $b_3^i$  are stored in *SRB2*. The  $3 \times$  multiplier, implemented by adding and shifting, generates the data  $3b_3^i$  and  $3b_2^i$ , which are stored in register *SRB3*. After that, the even coefficients of the ICT are generated from the data stored in *SRB1*, *SRB2*, and *SRB3*:  $Y_0^i$  and  $Y_2^i$  in *AE3* and  $Y_4^i$  and  $Y_6^i$  in *AE4*. The output mixer orders the even coefficient sequence of the ICT. In this processor, the adder *AE3* and the subtractor *AE4* have an efficiency of 100%.

### C. $\mathbf{J}_{40}$ Processor

The processor  $\mathbf{J}_{40}$  has been designed to calculate the odd coefficients of the 1-D  $\mathbf{J}$  transform. The implementation of this processor can be simplified through the decomposition of the  $\mathbf{J}_{40}$  matrix described in (17). This decomposition takes the form of the addition of the matrices whose elements are power of 2, resulting in

$$\begin{aligned} \begin{bmatrix} Y_1 \\ Y_3 \\ Y_5 \\ Y_7 \end{bmatrix} &= \begin{bmatrix} 2 & 2 & -2 & 2 \\ 2 & -2 & -2 & 2 \\ -2 & -2 & 2 & 2 \\ 2 & 2 & 2 & -2 \end{bmatrix} \begin{bmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \end{bmatrix} \\ &+ \begin{bmatrix} 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & -8 \\ 8 & 0 & 0 & 8 \\ 0 & -8 & 8 & 0 \end{bmatrix} \begin{bmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \end{bmatrix} \\ &- \begin{bmatrix} -8 & 1 & 0 & 0 \\ 1 & 0 & 8 & 0 \\ 0 & 8 & 0 & 1 \\ 0 & 0 & 1 & 8 \end{bmatrix} \begin{bmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \end{bmatrix}. \end{aligned} \quad (20)$$

In this form, the odd coefficients of the 1-D  $\mathbf{J}$  transform can be implemented simply in terms of add and shift operations. Fig. 5(a) shows the signal flow graph of  $\mathbf{J}_{40}$  obtained from (20). It has three computing stages with intermediate data  $d_i, f_i, e_i$ , and  $g_i$  ( $i = 0, 1, 2, 3$ ). Fig. 5(b) illustrates their architecture made up of five shift registers, ten multiplexers 4:1 and five arithmetic units operating in parallel. Fig. 5(c) shows the timing diagram of this processor. In this diagram, the input sequence  $\langle a_4^i, a_5^i, a_6^i, a_7^i \rangle$  stored in the register *SRA2* is processed in *AE5* and *AE6*, generating the intermediate data  $\langle d_0^i, d_1^i, d_2^i, d_3^i \rangle$ ;  $d_0^i$  and  $d_2^i$  that are stored in the register *SRD1*, whereas  $d_1^i$  and  $d_3^i$  are stored in *SRD2*. *AE7* and *AE8* generate the intermediate data  $\langle f_0^i, f_1^i, f_2^i, f_3^i \rangle$  in parallel;  $f_0^i$  and  $f_2^i$  are stored in the register *SRF1*, and  $f_1^i$  and  $f_3^i$  are stored in *SRF2*. Next, the intermediate data  $\langle e_0^i, e_1^i, e_2^i, e_3^i \rangle$  are calculated in *AE5* and *AE6* and stored in *SRD1* and *SRD2*.  $\langle g_0^i, g_1^i, g_2^i, g_3^i \rangle$  are calculated in parallel in *AE7* and *AE8* and stored in *SRF1* and *SRF2*. Finally, the output adder *AE9* generates, in natural order, the odd coefficients  $\langle Y_1^i, Y_3^i, Y_5^i, Y_7^i \rangle$  of the 1-D  $\mathbf{J}$  transform from  $\langle e_0^i, e_1^i, e_2^i, e_3^i \rangle$  and  $\langle g_0^i, g_1^i, g_2^i, g_3^i \rangle$ . Note in the timing diagram that all the

arithmetic elements of the processor  $\mathbf{J}_{40}$  also have an efficiency of 100%.

## IV. TWO-DIMENSIONAL ICT ARCHITECTURE

The 2-D ICT for a real input sequence  $\mathbf{x}$  is defined as

$$\mathbf{X} = \mathbf{T}_x \mathbf{T}^t = \mathbf{K} \mathbf{J}_x \mathbf{J}^t \mathbf{K}^t = \mathbf{K} \mathbf{Y} \mathbf{K} \quad (21)$$

where  $\mathbf{X}$ ,  $\mathbf{x}$ , and  $\mathbf{K}$  are  $8 \times 8$  matrices, and  $\mathbf{K}^t = \mathbf{K}$ . Reordering the input data and the transform coefficients applying the rules defined in (8) and (9), the 2-D ICT can be expressed as

$$\mathbf{X}' = \mathbf{T}_R \mathbf{x}' \mathbf{T}_R^t = \mathbf{K}_R \mathbf{Y}' \mathbf{K}_R. \quad (22)$$

According to (11), the 2-D  $\mathbf{J}$  transform is defined as

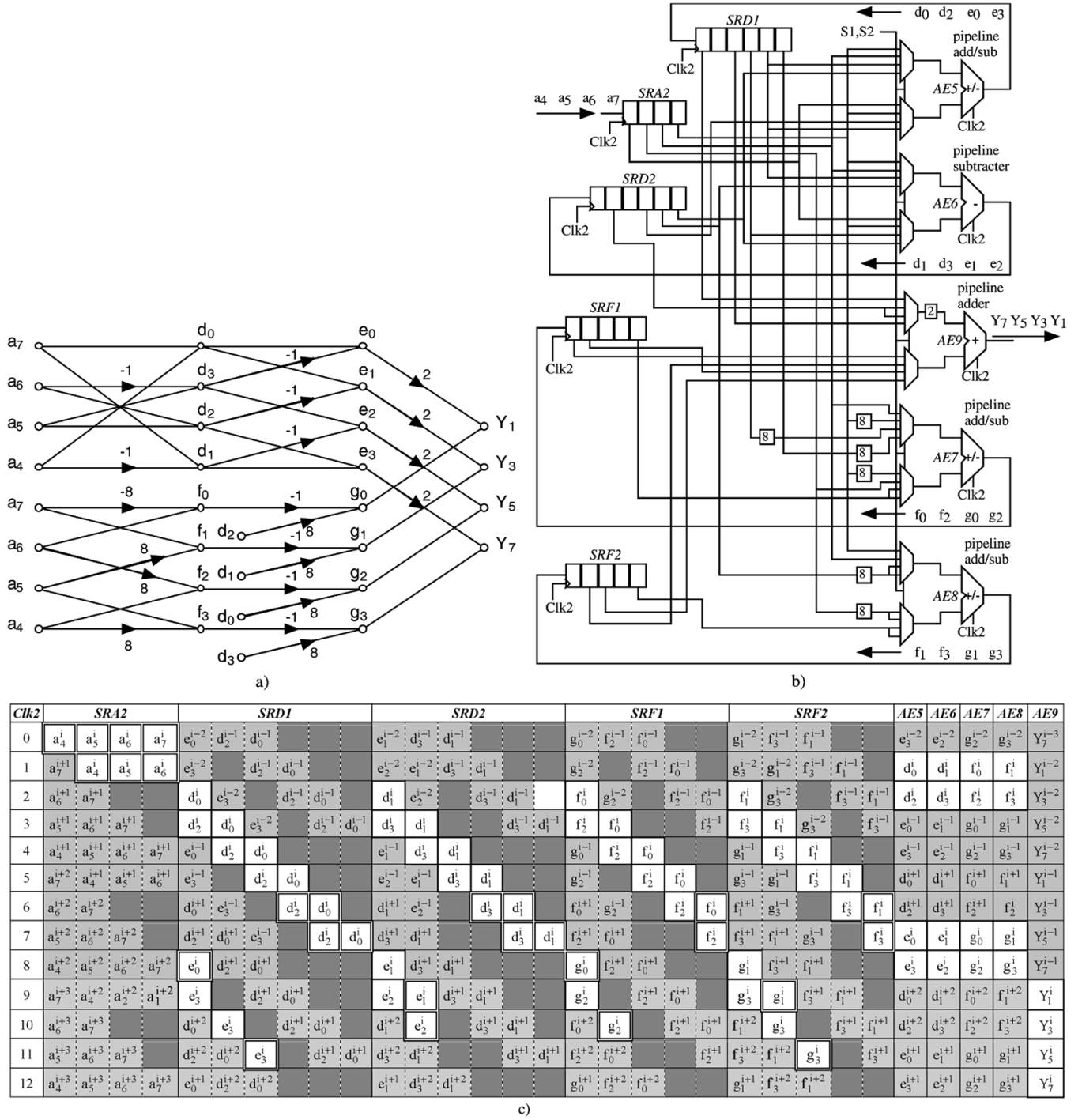
$$\mathbf{Y}' = \begin{bmatrix} \mathbf{J}_{4e} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{4o} \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \mathbf{x}' \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \begin{bmatrix} \mathbf{J}_{4e}^t & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{4o} \end{bmatrix}. \quad (23)$$

Fig. 6 shows the architecture of the  $8 \times 8$  2-D ICT processor, based on the computing scheme derived from (22) and (23). Because of the separability property, this 2-D  $\mathbf{J}$  transform is implemented using the two 1-D  $\mathbf{J}(10, 9, 6, 2, 3, 1)$  processors described above, with a transpose buffer as its intermediate memory. In this way, it adopts the so-called row-column method that performs 1-D  $\mathbf{J}$  operations on each row followed by another 1-D  $\mathbf{J}$  on each column. The output coefficients can be selected with or without normalization. In image and video processing, quantization is often required to compress the data. In this case, the final normalization of the transform is incorporated into the quantization step. For this purpose, the 2-D ICT processor allows both output formats, depending on its application.

### A. Transpose Buffer

The  $8 \times 8$  intermediate data generated by the first 1-D DCT processor has to be stored and transposed in the transpose buffer before the second 1-D DCT is performed. This transpose buffer allows simultaneous read and write operations between the two processors while doing matrix transposition. To achieve this, the data are read out of the buffer column-wise if the previous intermediate data were written into the buffer row-wise, and vice versa.

The transpose buffer based on D-type flip-flops has been chosen to be more adequate for pipeline architectures, unlike other proposed architectures based on RAM memories. A double buffering structure called the ‘‘ping-pong’’ buffer whose basic cells are two flip-flops that perform the transposing operation has been proposed in [26]. In this case, the data of registers are copied in parallel onto the corresponding adjacent registers that are connected in column-wise fashion. This circuit was subsequently improved by means of a transposition structure (TRAM) whose basic cell is a flip-flop and a multiplexer [27]. The TRAM has a parallel input/output structure and the data are transposed on the fly. The transpose buffer proposed for this processor is a variation of the previous TRAM with a more regular serial input/output structure. This circuit, whose scheme shown in Fig. 6, is made up of eight 16-bit shift-registers and

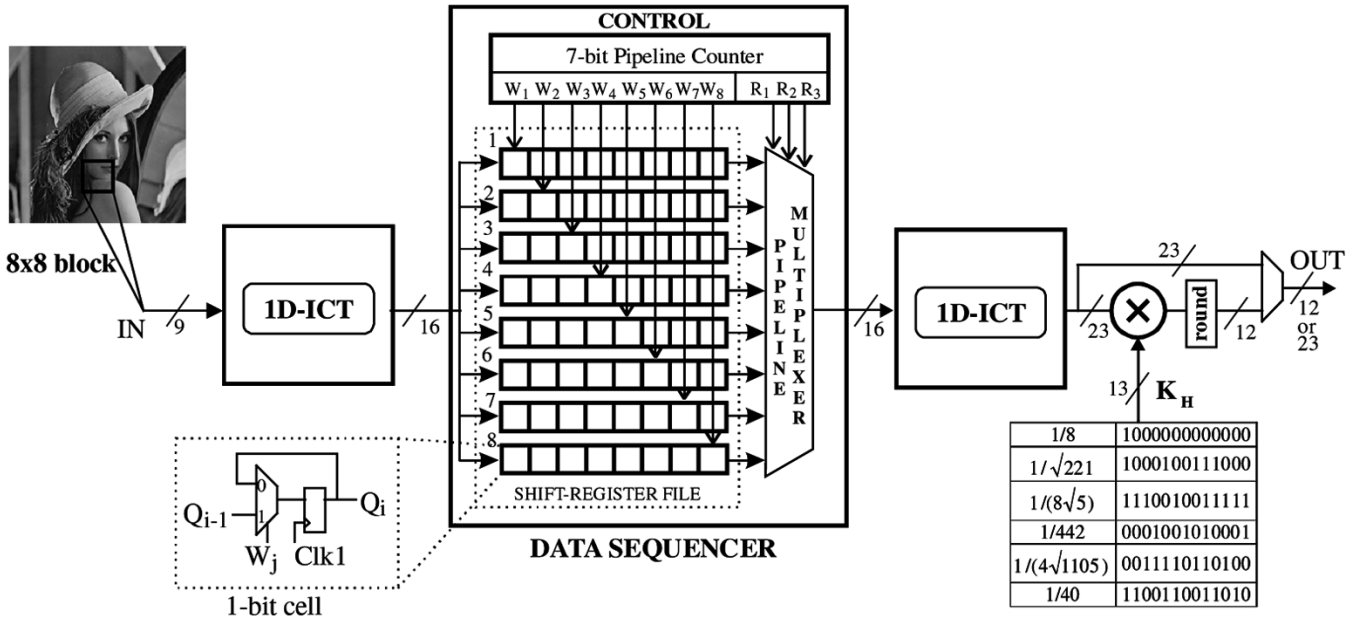

 Fig. 5.  $J_{40}$  processor: (a) Signal flow graph, (b) architecture, and (c) timing diagram.

a control circuit. The control circuit generates the signals  $W_j$  ( $j = 1$  to  $8$ ), which select one of the shift-registers independently, and  $R_k$  ( $k = 1$  to  $3$ ), which select, through a pipeline multiplexer, the output of the shift-register specified by  $W_j$ . In this way, the writing process, which is performed using  $W_j$ , and the reading process, performed using  $R_k$ , are synchronized in order to avoid losses of data. The control signals  $W_j$  and  $R_k$  are obtained from a 7-bit pipeline counter and some logic with a fine-grained pipeline structure, operating at the maximum frequency specified by  $Clk1$ . The problems associated with the high fan-out of the signals  $W_j$  and  $R_k$  have been solved by

combining parallel and tree buffering structures. The transpose buffer has a latency of 67 clock cycles.

### B. Accuracy Specifications

The DCT kernel components are real numbers so that truncation or rounding errors are inevitably introduced during computation. There are two inherent errors in a DCT implementation: 1) finite internal word length and 2) coefficients quantization error. The ICT is immune from this type 1) of error since its

Fig. 6. Scheme of  $8 \times 8$  2-D ICT chip image processing.

kernel components are integers. However, it presents type 2) errors in the final normalization of the transform. This normalization, expressed in (21), is carried out by transforming this equation in a Hadamard product ( $\cdot$ ) or element-by-element product of two matrices since the  $\mathbf{K}$  matrix is diagonal, giving

$$\mathbf{X} = \mathbf{K}_H \cdot \mathbf{Y} \quad (24)$$

where the normalization matrix  $\mathbf{K}_H$  is defined as (25), shown at the bottom of the page.

The coefficients word length of  $\mathbf{K}_H$  has been established through simulation with MATLAB using the specifications of IEEE standard 1180–1990 [35] as a reference. This standard specifies the numerical characteristics of the  $8 \times 8$  Inverse Discrete Cosine Transform (IDCT) for use in visual telephony and similar applications where the  $8 \times 8$  IDCT results are used in a reconstruction loop. Since its approval in 1990, much of the reported work on the IDCT implementation fulfills the precision specifications established in that document [8], [9]. The fulfillment of the specifications ensures the compatibility between different implementations of the IDCT. This standard has a set of very stringent requirements, but for video and image

applications where the IDCT is not used in a reconstruction loop, the requirements of the IDCT may be relaxed in order to lower its implementation costs. For those applications, the numeric characteristics of the forward discrete cosine transform (FDCT) may also be specified to guarantee picture quality [35].

A similar procedure to that proposed in [15] and shown in Fig. 7 was adopted to set up the coefficients' word length. The accuracy of the processor was calculated by comparing the results of a finite precision model of the proposed  $8 \times 8$  ICT architecture and an  $8 \times 8$  ICT 64-b floating-point reference model. Figs. 8 and 9 depict the maximum peak mean square error (MSE) (must be  $<0.06$ ) and overall MSE (OMSE) (must be  $<0.02$ ) for different coefficients' word length obtained at the FICT test output. Table I summarizes the accuracy specifications of the proposed ICT architecture defined in the standard. These results show that the minimum word length yielding compliance with reference specifications is 18-bit, of which only 13 bits are necessary for hardware implementation. Fig. 6 shows the binary representation of normalization coefficients and the  $23 \times 13$  multiplier to make this final normalization. This multiplier has a very fine-pipelined structure and includes a rounding circuit to the closest integer, and, as a result, the

$$\mathbf{K}_H = \begin{bmatrix} \frac{1}{8} & \frac{1}{4\sqrt{221}} & \frac{1}{8\sqrt{5}} & \frac{1}{4\sqrt{221}} & \frac{1}{8} & \frac{1}{4\sqrt{221}} & \frac{1}{8\sqrt{5}} & \frac{1}{4\sqrt{221}} \\ \frac{1}{4\sqrt{221}} & \frac{1}{442} & \frac{1}{4\sqrt{1105}} & \frac{1}{442} & \frac{1}{4\sqrt{221}} & \frac{1}{442} & \frac{1}{4\sqrt{1105}} & \frac{1}{442} \\ \frac{1}{8\sqrt{5}} & \frac{1}{4\sqrt{1105}} & \frac{1}{40} & \frac{1}{4\sqrt{1105}} & \frac{1}{8\sqrt{5}} & \frac{1}{4\sqrt{1105}} & \frac{1}{40} & \frac{1}{4\sqrt{1105}} \\ \frac{1}{4\sqrt{221}} & \frac{1}{442} & \frac{1}{4\sqrt{1105}} & \frac{1}{442} & \frac{1}{4\sqrt{221}} & \frac{1}{442} & \frac{1}{4\sqrt{1105}} & \frac{1}{442} \\ \frac{1}{8} & \frac{1}{4\sqrt{221}} & \frac{1}{8\sqrt{5}} & \frac{1}{4\sqrt{221}} & \frac{1}{8} & \frac{1}{4\sqrt{221}} & \frac{1}{8\sqrt{5}} & \frac{1}{4\sqrt{221}} \\ \frac{1}{4\sqrt{221}} & \frac{1}{442} & \frac{1}{4\sqrt{1105}} & \frac{1}{442} & \frac{1}{4\sqrt{221}} & \frac{1}{442} & \frac{1}{4\sqrt{1105}} & \frac{1}{442} \\ \frac{1}{8\sqrt{5}} & \frac{1}{4\sqrt{1105}} & \frac{1}{40} & \frac{1}{4\sqrt{1105}} & \frac{1}{8\sqrt{5}} & \frac{1}{4\sqrt{1105}} & \frac{1}{40} & \frac{1}{4\sqrt{1105}} \\ \frac{1}{4\sqrt{221}} & \frac{1}{442} & \frac{1}{4\sqrt{1105}} & \frac{1}{442} & \frac{1}{4\sqrt{221}} & \frac{1}{442} & \frac{1}{4\sqrt{1105}} & \frac{1}{442} \end{bmatrix} \quad (25)$$

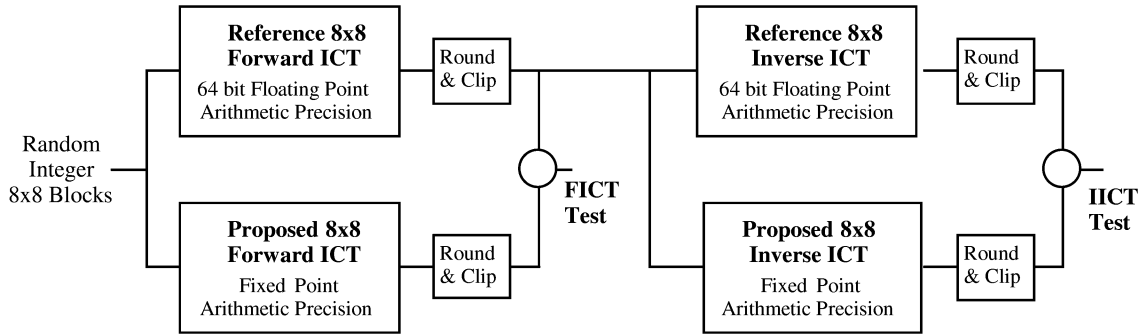


Fig. 7. Procedure for ICT accuracy measurement.

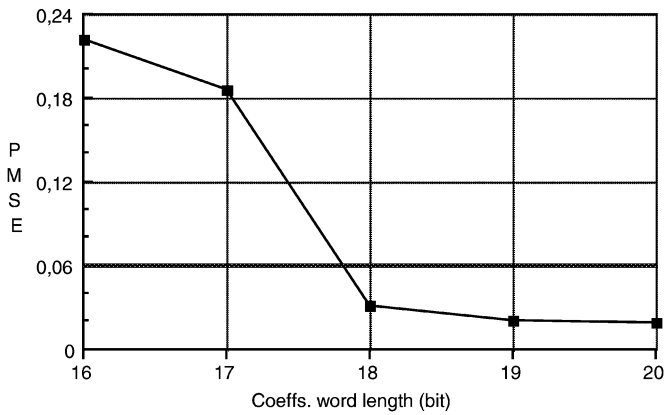


Fig. 8. Maximum peak mean square (PMSE) error as function of the normalization coefficients word length.

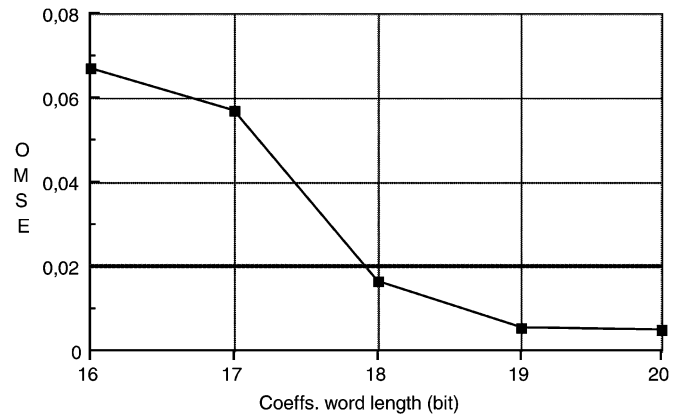


Fig. 9. OMSE as function of the normalization coefficients word length.

normalized output data of the 2-D ICT processor are coded as two's complement integers of 12 bits.

Fig. 7 also shows the accuracy measurement process of the inverse ICT made by simulation using MATLAB. Similar results of accuracy specifications are obtained in comparison with the forward ICT, and 18-bit (13-bit effective) is the minimum word length of coefficients. This is because the truncation and round errors only take place in the quantification of the normalization coefficients.

### C. Chip Implementation and Comparisons

A prototype of an  $8 \times 8$  2-D ICT processor chip for image transform has been designed using standard cells in a semi-custom methodology. The processor was implemented with a  $0.35\text{-}\mu\text{m}$  CMOS CSD 3M/2P 3.3-V technology from Austria-Microsystem [33]. The chip has an area of  $2900 \times 200 \approx 9.3\text{ mm}^2$  (the core is  $2200 \times 2480 \approx 5.5\text{ mm}^2$ ), where the multiplier and the transpose buffer account for nearly 50% of the total surface. It contains 12.4 k gates, 6.7 k gates of which are flip-flops. The chip was encapsulated on a JLCC 68 package with the following distribution of pads: 12 input, 26 output, 11 GND, and 11 VDD. It was fully tested and a maximum operating frequency of about 300 MHz has been established. It needs 214 Clk1 cycles to complete a 2-D transform without normalization and 260 cycles with normalization. The time to compute a 2-D order-8 ICT transform is close to 710 (without normalization) and 860 ns (with normalization).

Table II lists features of the proposed processor and other ICT and DCT hardware implementations. The 2-D ICT(10,9,6,2,9,3) processor described in [28] is made up of three chips: two 1-D ICT processors and a data sequencer, each one implemented on a  $3\text{-}\mu\text{m}$  gate array with 2800 gates. This processor computes the 2-D order-8 ICT in less than 19  $\mu\text{s}$  at 3 MHz clock rate. It uses only three arithmetic units: an adder/subtractor, a multiplier based on shift-add algorithm, and an ALU. The 1-D ICT(10,9,6,2,9,3) chip proposed in [29] has been designed using a self-timed technique in  $0.7\text{-}\mu\text{m}$  CMOS technology combining standard cells from ES21 library with a few full-custom cells. The data input rate is up to 50 MHz at  $V_{DD} = 5\text{ V}$ , and their average performance is 142.1 ns and 173.2 ns for forward and inverse transform, respectively. It uses an integer execution unit made up of a multiplier based on multiplexers and adders to perform  $\times 1/2$ ,  $\times 2$ ,  $\times 3$ ,  $\times 4$ , and  $\times 5$  operations and an adder/subtractor. For proposes of comparison, Table II also includes a selection of DCT processors with technologies similar to those used in this ICT processor. The proposed ICT is clearly superior to other two ICT processors in terms of speed, power, and area, despite the difference in technology. The parallel-pipeline architecture and the arithmetic units operating at half the frequency allows an input data rate of 300 MHz, which is the fastest presented up to now. The speed of ICT processor is faster than the DCT processors, and its architecture is efficient-area in terms of transistor number. However, an efficient full-custom design of the transpose buffer in place of the D-type flip-flops and multiplexer standard cells would allow a saving of up to 80% in



TABLE I  
ACCURACY TEST RESULT OF THE ICT PROCESSOR

Input range	PE ( $1^2 < 1$ )	IPMEI ( $< 0.015$ )	PMSE ( $< 0.06$ )	IOMEI ( $< 0.0015$ )	OMSE ( $< 0.02$ )
[-5,5]	[-1,1]	0.0022	0.0294	0.000044	0.00223
[5,-5]	[-1,1]	0.0032	0.0264	0.000044	0.002
[-256,255]	[-1,1]	0.0032	0.0318	0.000023	0.01643
[255,-256]	[-1,1]	0.0025	0.0338	0.000116	0.01639
[-300,300]	[-1,1]	0.0057	0.0373	0.000097	0.01915
[300,-300]	[-1,1]	0.0038	0.0381	0.000342	0.01875

PE: peak error, PME: peak mean error, PMSE: peak mean square error, OME: overall mean error, OMSE: overall mean square error.

TABLE II  
PROCESSOR COMPARISON

	ICT			DCT				
	[28]	[29]	Ours	[5]	[6]	[11]	[13]	[14]
<b>Function</b>	2-D ICT	1-D ICT	2-D ICT	2-D DCT	2-D DCT	2-D DCT	2-D DCT	2-D IDCT
<b>Technology</b>	3 $\mu$ m CMOS gate array	0.7 $\mu$ m CMOS full/semi custom	0.35 $\mu$ m CMOS semi-custom	0.6 $\mu$ m CMOS semi-custom	0.3 $\mu$ m CMOS full custom	0.6 $\mu$ m CMOS semi-custom	0.35 $\mu$ m CMOS semi-custom	0.6 $\mu$ m CMOS semi-custom
<b>Chip size</b>	3 chips	23.4 mm <sup>2</sup>	9.3 mm <sup>2</sup>	30.3 mm <sup>2</sup>	4 mm <sup>2</sup>	70 mm <sup>2</sup>	18.2 mm <sup>2</sup>	12.3 mm <sup>2</sup>
<b>Transistors/gates</b>		76k trans.	12.4k gates (estim. 70k trans.)	39k gates 156k trans.	120k trans.	152k trans.	119k trans.	78k trans.
<b>Supply voltage</b>		5 V	3.3 V		0.9 V	2.0 V	3.3 V	3.3 V
<b>Power dissipation</b>		310mW@50 MHz	170mW@300M Hz		10mW@150 MHz	133mW@100M Hz		120mW@100M Hz
<b>Frequency</b>	3 MHz	50/60 MHz	300 MHz	100 MHz	150 MHz	133MHz	100 MHz	100 MHz
<b>Latency</b>	19 $\mu$ s	142.1/173.2ns	214/260 clocks		112 clocks	198 clocks		86 clocks

area. Moreover, the output multiplier can be incorporated to the decoding process in an adaptive transform coding system such as that proposed in [21]. As the two circuits occupy almost 50% of the total area, the full-custom design of this processor would be highly area-efficient and would allow far higher frequencies to be achieved than those obtained with the standard cell design methodology used here.

## V. CONCLUSION

This paper presents an  $8 \times 8$  forward 2-D ICT(10, 9, 6, 2, 3, 1) processor for image encoding. This processor was manufactured using a 0.35- $\mu$ m CMOS process on a 9.3-mm<sup>2</sup> die. It includes 12.4 k gates (6757 of which are flip-flops), and it has been conceived using a standard cell design methodology. The 2-D ICT architecture is made up of two 1-D ICT(10, 9, 6, 2, 3, 1) processors and a transpose buffer used as intermediate memory. The parallel-pipeline architecture proposed allows a speed of 300 MHz, and the pipelined adders/subtractors have 100% efficiency operating at half the frequency of the input data rate. Other characteristics of this architecture are high throughput, parallel processing, and reduced internal storage. The authors think the ICT will be used in increasingly many applications in the future.

## ACKNOWLEDGMENT

The authors wish to thank the anonymous referees for their suggestions and remarks, which enabled us to improve the manuscript.

## REFERENCES

- [1] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, and Applications*. New York: Academic, 1990.
- [2] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architectures*, Second ed. Boston, MA: Kluwer, 1997.
- [3] K. R. Rao and J. J. Hwang, *Techniques and Standards for Image Video and Audio Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [4] C. L. Wang and C. Y. Chen, "High-throughput VLSI architectures for the 1-D and 2-D discrete cosine transforms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 1, pp. 31–40, Feb. 1995.
- [5] K. H. Cheng, C. S. Huang, and C. P. Lin, "The design and implementation of DCT/IDCT chip with novel architecture," in *Proc. IEEE Int. Symp. Circuits Syst.*, Geneva, Switzerland, May 28–31, 2000, pp. IV-741–IV-744.
- [6] T. Kuroda, T. Fujita, S. Mita, T. Nagamatsu, S. Yoshioka, K. Suzuki, 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<sup>2</sup>, 2-D discrete cosine transform core processor with variable threshold-voltage (VT) scheme," *IEEE J. Solid-State Circuits*, vol. 31, no. 11, pp. 1770–1779, Nov. 1996.
- [7] G. Bi, G. Li, K. K. Ma, and T. C. Tan, "On the computation of two-dimensional DCT," *IEEE Trans. Signal Process.*, vol. 48, no. 4, pp. 1171–1183, Apr. 2000.
- [8] P. C. Jain, W. Schlenk, and M. Riegel, "VLSI implementation of two-dimensional DCT processor in real time for video codec," *IEEE Trans. Consum. Electron.*, vol. 38, no. 3, pp. 537–545, Aug. 1992.
- [9] S. Kim and W. Sung, "Optimum wordlength determination of  $8 \times 8$  IDCT architectures conforming to the IEEE standard specifications," in *Conf. Rec. Twenty Ninth Asilomar Conf. Signals, Syst., Comput.*, vol. 2, 1996, pp. 821–5.
- [10] H. C. Chang, J. J. Jiu, L. L. Chen, and L. G. Chen, "A low power  $8 \times 8$  direct 2-D DCT chip design," in *J. VLSI Signal Processing Syst. Signal, Image, Video Technol.*, vol. 26, Nov. 2000, pp. 319–332.
- [11] L. G. Chen, J. Y. Jiu, H. C. Chang, Y. P. Lee, and C. W. Ku, "A low power  $8 \times 8$  direct 2-D DCT chip design," in *J. VLSI Signal Process.*, vol. 26, 2000, pp. 319–332.
- [12] S. F. Hsiao and J. M. Tseng, "Parallel, pipelined, and folded architectures for computation of 1-D and 2-D DCT in image and video codec," in *J. VLSI Signal Process. Syst. Signal, Image, Video Technol.*, vol. 28, July 2001, pp. 205–220.
- [13] J. S. Chiang, Y. F. Chiu, and T. H. Chang, "A high throughput 2-dimensional DCT/IDCT architecture for real-time image and video system," in *Proc. 8th IEEE Int. Conf. Electron., Circuits, Syst.*, vol. 2, Piscataway, NJ, 2001, pp. 867–870.
- [14] T. H. Chen, "A cost-effective  $8 \times 8$  2-D IDCT core processor with folded architecture," *IEEE Trans. Consum. Electron.*, vol. 45, no. 2, pp. 333–339, May 1999.

- [15] I.-K. Kim, J.-J. Cha, and H.-J. Cho, "A design of 2-D DCT/IDCT for real-time video applications," in *Proc. ICVC, 6th Int. Conf. VLSI CAD*, 1999, pp. 557–559.
- [16] Y. Zeng, L. Cheng, G. Bi, and A. C. Kot, "Approximation of DCT without multiplication in JPEG," in *Proc. 3rd IEEE Int. Conf. Electron., Circuits Syst.*, vol. 2, 1996, pp. 704–707.
- [17] N. Merhav and V. Bhaskaran, "A multiplication-free approximate algorithm for the inverse discrete cosine transform," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, 1999, pp. 759–763.
- [18] T. D. Tran, "The BinDCT: Fast multiplierless approximation of the DCT," *IEEE Signal Processing Lett.*, vol. 7, no. 6, pp. 141–4, Jun. 2000.
- [19] J. Liang and T. D. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," *IEEE Trans. Signal Process.*, vol. 49, no. 12, pp. 3032–3044, Dec. 2001.
- [20] Y. Zeng, L. Cheng, G. Bi, and A. C. Kot, "Integer DCT's and fast algorithms," *IEEE Trans. Signal Process.*, vol. 49, no. 11, pp. 2774–2784, Nov. 2001.
- [21] W. K. Cham, "Development of integer cosine transforms by the principle of dyadic symmetry," in *Proc. Inst. Elect. Eng. I*, vol. 136, Aug. 1989, pp. 276–282.
- [22] W. K. Cham and Y. T. Chan, "An order-16 integer cosine transform," *IEEE Trans. Signal Process.*, vol. 39, no. 5, pp. 1205–1208, May 1991.
- [23] F. S. Wu and W. K. Cham, "A comparison of error behavior in the implementation of the DCT and the ICT," in *Proc. IEEE Region 10 Conf. Comput. Commun. Syst.*, Sep. 1990, pp. 450–453.
- [24] A. Marcek, J. Kotuliakova, and G. Rozinaj, "New approach of fast ICT and MICT algorithms development," in *Proc. 3rd IEEE Int. Conf. Electron., Circuits, Syst.*, 1996, pp. 744–747.
- [25] M. Costa and K. Tong, "A Simplified Integer Cosine Transform and Its Application in Image Compression," TDA Progress Rep. 42-119, Nov. 15, 1994. NASA Code 310-30-71-83-04.
- [26] M. Kovac and N. Ranganathan, "JAGUAR: A fully pipelined VLSI architecture for JPEG image compression standard," *Proc. IEEE*, vol. 83, no. 2, pp. 247–258, Feb. 1995.
- [27] T. Xanthopoulos and A. P. Chandrakasan, "A low-power IDCT macrocell for MPEG-2 MP@ML exploiting data distribution properties for minimal activity," *IEEE J. Solid State Circuits*, vol. 34, no. 5, pp. 693–703, May 1999.
- [28] W. K. Cham, C. S. O. Choy, and W. K. Lam, "A 2-D integer cosine transform chip set and its applications," *IEEE Trans. Consum. Electron.*, vol. 38, no. 2, pp. 43–47, May 1992.
- [29] T. C. J. Pang, C. S. O. Choy, C. F. Chan, and W. K. Cham, "A self-timed ICT chip for image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 6, pp. 856–860, Sep. 1999.
- [30] Y. J. Chen, S. Orintara, and T. Nguyen, "Video compression using integer DCT," in *Proc. Int. Conf. Image Processing*, vol. 2, 2000, pp. 844–847.
- [31] G. A. Ruiz, J. A. Michell, A. M. Burón, J. M. Solana, M. A. Manzano, and F. J. Díaz, "Integer cosine transform chip design for image compression," in *Proc SPIE First Int. Symp. Microtechnologies New Millennium: VLSI Circuits Syst.*, vol. 5117, Maspalomas, Gran Canaria, Spain, May 2003, pp. 33–41.
- [32] A. Michell, G. A. Ruiz, J. Liang, and A. M. Burón, "Parallel-pipelined architecture for 2-DICT VLSI implementation," in *Proc. IEEE Int. Conf. Image Process.*, Barcelona, Spain, Sep. 14–17, 2003, pp. III-89–III-92.
- [33] [Online]. Available: <http://asic.austriamicrosystems.com>
- [34] R. P. Brent and H. T. Kung, "A regular layout for parallel adders," *IEEE Trans. Comput.*, vol. C-31, no. 3, pp. 260–264, Mar. 1982.
- [35] *IEEE Standard Specifications for the Implementations of  $8 \times 8$  Inverse Discrete Cosine Transform*, Mar. 1991. New York.



**Gustavo A. Ruiz** was born in Burgos, Spain, in 1962. He received the M.Sc. degree in physics in 1985 from the University of Navarra, Pamplona, Spain, and the Ph.D. degree in physical science in 1989 from the University of Cantabria, Santander, Spain.

Since 1985, he has been with the Department of Electronics and Computers, University of Cantabria, where he is currently an Associate Professor. His current research interests are mainly focused on VLSI architectures for signal processing and high-speed arithmetic circuits.



**Juan A. Michell** was born in Cáceres, Spain, in 1952. He received the M.S. and the Ph.D. degrees in physical sciences from the University of Cantabria, Santander, Spain, in 1974 and 1977, respectively.

Since 1974, he has been with the Department of Electronics and Computers, University of Cantabria, where he was appointed Professor in electronics in 1991. His current research interests are VLSI architectures and integrated circuit design for digital signal processing applications.



**Angel M. Burón** was born in Córdoba, Spain, in 1948. He received the M.S. and the Ph.D. degrees in physical sciences from the University of Sevilla, Seville, Spain, in 1969 and 1973, respectively.

Since 1973, he has been with the Department of Electronics and Computers at the University of Cantabria, Santander, Spain, where he was appointed Professor of electronics in 1982. His current research interests are integrated circuit design and VLSI/ULSI architectures for digital signal processing.