An integer cosine transform chip design for image compression

G. A. Ruiz, J. A. Michell, A. M. Burón, J. M. Solana, M. A. Manzano, J. Díaz
Department of Electronic and Computers, Facultad de Ciencias, University of Cantabria
Avda.Los Castros s/n. 39005-Santander. SPAIN
michellj@unican.es

ABSTRACT

The Integer Cosine Transform (ICT) has been shown to be an alternative to the Discrete Cosine Transform (DCT) for image processing. This paper presents a parallel-pipelined architecture of an 8x8 ICT(10, 9, 6, 2, 3, 1) processor for image compression. The main characteristics of this architecture are: high throughput, low latency, reduced internal storage and 100% efficiency in all computational elements. The processor has been designed in 0.35-µm CMOS technology with an estimated operational frequency of 300MHz.

Keywords: Integer cosine transform, multiplication free DCT, discrete cosine transform, image compression, parallel pipelined architectures, VLSI.

INTRODUCTION

Since the introduction of the Discrete Cosine Transform (DCT) several contributions have been published on the subject of fast algorithms and architectures for different applications, many of which describe 2D-DCT processors for image compression [1-6] and approximations to the DCT that are multiplication-free in order to reduce implementation complexity [7-8]. Although the DCT is the most widely used transform for image processing, other transforms have appeared in the last two decades which have less compression capacity and lower implementation costs [9-13]. From them, the Integer Cosine Transform (ICT) has been shown to be a promising alternative to the DCT due to its implementation simplicity, similar performance and compatibility with the DCT [9]. The ICT(10, 9, 6, 2, 3, 1) has the advantage of not needing multipliers, as its kernel is a matrix of integers [12-13].

This paper describes the design and implementation of an 8x8 2-D ICT processor for image compression, which meets the numerical characteristic of the IEEE std. 1180-1990. This processor uses a low latency data flow that minimizes the internal memory and a parallel pipelined architecture, based on a numerical strength reduction Integer Cosine Transform (10, 9, 6, 2, 3,1) algorithm, in order to attain high throughput and continuous data flow. A prototype of the 8x8 ICT processor has been implemented using a standard cell design methodology and a 0.35-µm CMOS CSD 3M/2P 3.3V process [14] on a 10mm^2 die. Pipeline circuit techniques have been used to attain the frequency of operation of 300MHz, maximum allowed by the technology. The circuit includes 12446 cells, 6757 of which are flip-flops. Two clock signals have been distributed: an external one, Clk1, operating at input data sampling frequency f_s, and an internal one, Clk2, operating at $f_s/2$. The high number of flip-flops has forced the use of a strategy to minimize clock-skew, combining large sized buffers on the periphery and using wide metal lines (clock-trunks) to distribute the signals.

2. DECOMPOSITION OF THE INTEGER COSINE TRANSFORM

The ICT was derived from the DCT through the concept of dyadic symmetry. The order-8 ICT kernel is

$$\mathbf{T} = \mathbf{K}\,\mathbf{J} \tag{1}$$

where **K** is the normalization matrix, and **J** an orthogonal matrix defined as

$$\mathbf{J}=\begin{bmatrix} g & g & g & g & g & g & g & g \\ a & b & c & d & -d & -c & -b & -a \\ e & f & -f & -e & -e & -f & f & e \\ b & -d & -a & -c & c & a & d & -b \\ g & -g & -g & g & g & -g & -g & g \\ c & -a & d & b & -b & -d & a & -c \\ f & -e & e & -f & -f & e & -e & f \\ d & -c & b & -a & a & -b & c & -d \end{bmatrix} \qquad (2)$$

whose elements are all integers and satisfy

$$a\,b = a\,c + b\,d + c\,d \quad , \quad a \geq b \geq c \geq d \quad \text{and} \quad e \geq f \qquad (3)$$

There are many possible **J** matrices, and the corresponding ICTs are denoted as ICT(a, b, c, d, e, f); g is always 1.

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

$$\mathbf{X} = \mathbf{T}\,\mathbf{x} = \mathbf{K}\,\mathbf{J}\,\mathbf{x} = \mathbf{K}\,\mathbf{Y} \qquad (4)$$

where **X** and **x** are dimension-8 column matrices. Reordering the input sequence and the transform coefficients according to the rules

$$x'(n) = x(n) \quad , \quad x'(7-n) = x(n+4) \quad , \quad n \in [0,3] \qquad (5)$$

$$X'(m) = X(Br8[m]) \quad , \quad X'(m+4) = X(2m+1) \quad , \quad m \in [0,3] \qquad (6)$$

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

$$\mathbf{X'} = \mathbf{T}_R\,\mathbf{x'} = \mathbf{K}_R\,\mathbf{J}_R\,\mathbf{x'} = \mathbf{K}_R\,\mathbf{Y'} \qquad (7)$$

The reordered integer ICT kernel is

$$\mathbf{J}_R = \begin{bmatrix} \mathbf{J}_{4e} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{4o} \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & \mathbf{-I}_4 \end{bmatrix} \qquad (8)$$

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

$$\mathbf{J}_{4e} = \begin{bmatrix} g & g & g & g \\ g & -g & -g & g \\ e & f & -f & -e \\ f & -e & e & -f \end{bmatrix} \quad , \quad \text{and} \quad \mathbf{J}_{4o} = \begin{bmatrix} a & b & c & d \\ b & -d & -a & -c \\ c & -a & d & b \\ d & -c & b & -a \end{bmatrix} \qquad (9)$$

Applying the decomposition rules defined in equations (5) and (6) to the $\mathbf{J}_{4e}$ matrix results

$$\mathbf{J}_{4e} = \begin{bmatrix} \mathbf{J}_{2e} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{2o} \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 \\ \mathbf{I}_2 & \mathbf{-I}_2 \end{bmatrix} \mathbf{R}_4 \qquad (10)$$

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

$$\mathbf{J}_{2e} = \begin{bmatrix} g & g \\ g & -g \end{bmatrix} \quad \text{and} \quad \mathbf{J}_{2o} = \begin{bmatrix} e & f \\ f & -e \end{bmatrix} \qquad (11)$$

Figure 1 shows the signal flow graph obtained by applying the decomposition process to **J**(10, 9, 6, 2, 3, 1). In this case

$$\mathbf{J}_{4e}=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & -1 & 1\\ 3 & 1 & -1 & -3\\ 1 & -3 & 3 & -1\end{bmatrix},\ \text{and}\ \mathbf{J}_{4o}=\begin{bmatrix}10 & 9 & 6 & 2\\ 9 & -2 & -10 & -6\\ 6 & -10 & 2 & 9\\ 2 & -6 & 9 & -10\end{bmatrix} \tag{12}$$

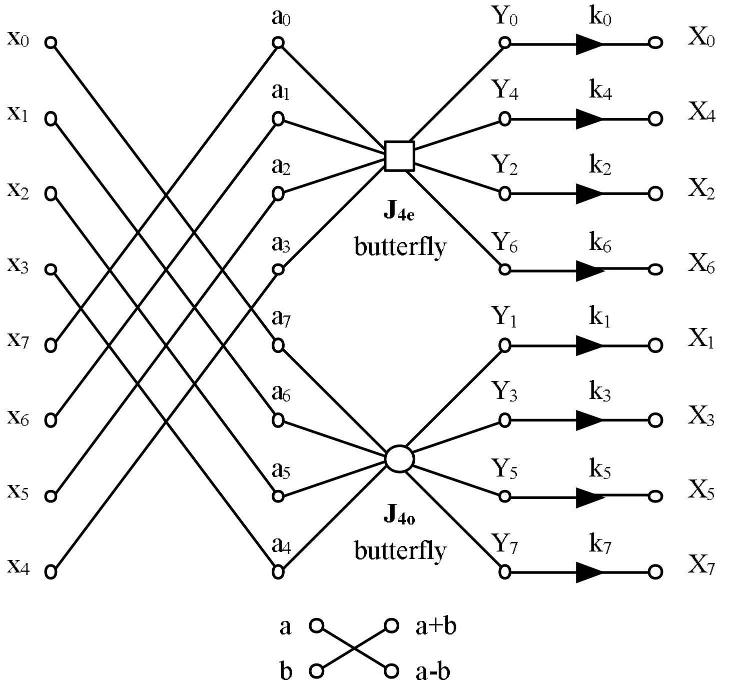

Figure 1. Signal flow graph of 1-D ICT

As can be seen in Figure 1, the first computing level operates on the input data, reordering them according to rule (5); additions and subtractions of data pairs formed with sequences x'(n) and x'(n+4) (n= 0, 1, 2, 3) are executed. In the second computing level the transformations $\mathbf{J}_{4e}$ and $\mathbf{J}_{4o}$, are obtained, their nuclei being the matrices defined by (12). The transformation $\mathbf{J}_{4e}$ is applied to the first half of the intermediate data sequence, a_0 to a_3, giving as a result the even coefficients $<Y_0, Y_2, Y_4, Y_6>$ of the ICT, without normalization. Similarly, $\mathbf{J}_{4o}$ is applied to the other half of the middle data sequence, a_7 to a_4, giving as a result the odd coefficients $<Y_1, Y_3, Y_5, Y_7>$ of the ICT, also without normalization. In the third computing level, the coefficients Y_i are normalized by k_i and the transform sequence of the coefficients X(m) appears reordered according to rule (6). Applying the decomposition procedure of $\mathbf{J}_{4e}$ established in (10) and (11), we get:

$$\begin{bmatrix}Y_0\\ Y_4\\ Y_2\\ Y_6\end{bmatrix}=\begin{bmatrix}1 & 1 & 0 & 0\\ 1 & -1 & 0 & 0\\ 0 & 0 & 3 & 1\\ 0 & 0 & 1 & -3\end{bmatrix}\begin{bmatrix}1 & 0 & 1 & 0\\ 0 & 1 & 0 & 1\\ 1 & 0 & -1 & 0\\ 0 & 1 & 0 & -1\end{bmatrix}\begin{bmatrix}a_0\\ a_1\\ a_3\\ a_2\end{bmatrix}=\begin{bmatrix}1 & 1 & 0 & 0\\ 1 & -1 & 0 & 0\\ 0 & 0 & 3 & 1\\ 0 & 0 & 1 & -3\end{bmatrix}\begin{bmatrix}b_0\\ b_1\\ b_3\\ b_2\end{bmatrix} \tag{13}$$

b_0, b_1, b_2 and b_3 being the intermediate data of the computation of transformation $\mathbf{J}_{4e}$. Operating on (13), we get:

$$\begin{bmatrix}Y_0\\ Y_4\end{bmatrix}=\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}b_0\\ b_1\end{bmatrix}\quad \text{and}\quad \begin{bmatrix}Y_2\\ Y_6\end{bmatrix}=\begin{bmatrix}3 & 1\\ 1 & -3\end{bmatrix}\begin{bmatrix}b_3\\ b_2\end{bmatrix} \tag{14}$$

Figure 2 shows the signal flow graph obtained from (13) and (14).

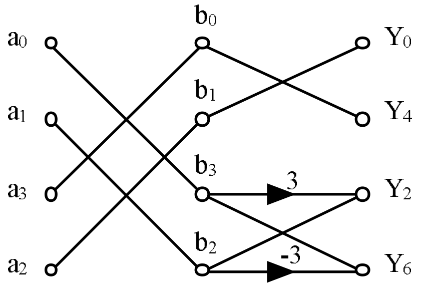

Figure 2. Signal flow graph of transform $\mathbf{J}_{4e}$

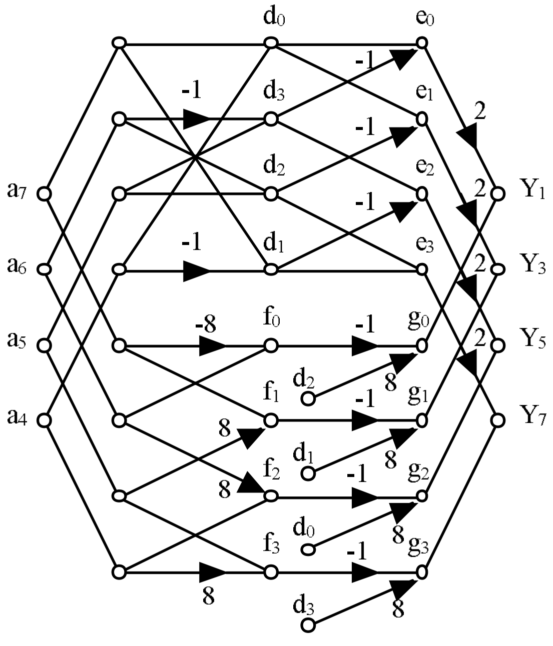

Figure 3. Signal flow graph of transform $\mathbf{J}_{4o}$

As can be seen in (13), the computation of the even coefficients of the ICT can be performed with additions and subtractions, as multiplication by 3 can be easily implemented by means of add and shift operations. The computation of the odd coefficients of the ICT can also be simplified; decomposition of the $\mathbf{J}_{4o}$ matrix as the addition of matrices having elements that are powers of 2, gives:

$$\begin{bmatrix} Y_1 \\ Y_3 \\ Y_5 \\ Y_7 \end{bmatrix} = \begin{bmatrix} 2 & 2 & -2 & 2 \\ 2 & -2 & -2 & 2 \\ -2 & -2 & 2 & 2 \\ 2 & 2 & 2 & -2 \end{bmatrix} \begin{bmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \end{bmatrix} + \begin{bmatrix} 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & -8 \\ 8 & 0 & 0 & 8 \\ 0 & -8 & 8 & 0 \end{bmatrix} \begin{bmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \end{bmatrix} - \begin{bmatrix} -8 & 1 & 0 & 0 \\ 1 & 0 & 8 & 0 \\ 0 & 8 & 0 & 1 \\ 0 & 0 & 1 & 8 \end{bmatrix} \begin{bmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \end{bmatrix} \qquad (15)$$

In this form, the odd coefficients of the ICT can be implemented simply in terms of add and shift operations. Figure 3 shows the signal flow graph, obtained from (15) for the transformation $\mathbf{J}_{4o}$, having three computing levels, d_i, f_i, e_i and g_i (i=0, 1, 2, 3) being the intermediate data.

3. 1-D ICT PARALLEL PIPELINED ARCHITECTURE

The 1-D ICT processor architecture, whose scheme is shown in Figure 4, has been designed to implement the computing diagram of Figure 1 with the highest degree of efficiency. It has an input processor computing the intermediate data of the first computing level, a_0 to a_7, two processors in parallel, computing the transformations $\mathbf{J}_{4e}$ and $\mathbf{J}_{4o}$, and an output mixer generating the coefficients sequence of the ICT, ordered in natural form. The 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 output mixer gives the coefficients sequence of the ICT at the frequency f_s.

The input processor (Fig. 4) has a shift register of length 11 which stores the input data sampled at frequency f_s, two multiplexers 4:1, an adder and a subtracter. The adder and the subtracter both have pipeline structure and operate in parallel at frequency $f_s/2$, generating the input sequences, $<a_0, a_1, a_2, a_3>$ and $<a_7, a_6, a_5, a_4>$, of processors $\mathbf{J}_{4e}$ and $\mathbf{J}_{4o}$ from the data stored in the register. In this way an efficiency of 100% is attained for the arithmetic elements.

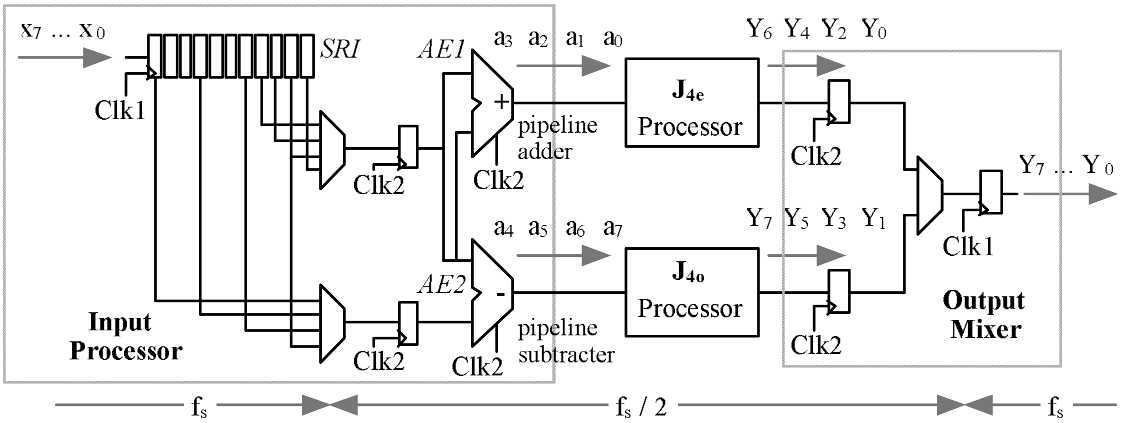

Figure 4. Architecture of processor 1D ICT

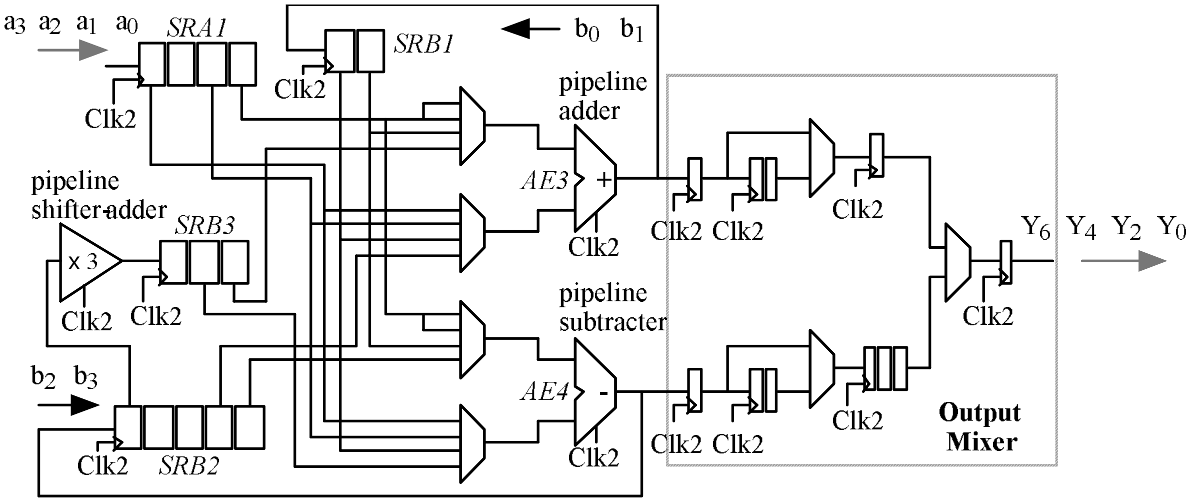

Figure 5. Architecture of processor $\mathbf{J_{4e}}$

The processor $\mathbf{J_{4e}}$ has been conceived to calculate the even coefficients of the ICT using the procedure established in (13) and in the signal flow graph of Figure 2. As can be seen in Figure 5, this processor has four shift registers, four multiplexers 4:1, three arithmetic units having pipeline structure, and an output mixing and ordering circuit. The adder and the subtracter, which operate in parallel at frequency $f_s/2$, generate first the intermediate data $<b_0, b_1, b_2, b_3>$ from the input sequence $<a_0, a_1, a_2, a_3>$, stored in the register *SRA1*; b_0 and b_1 are stored in register *SRB1*, while b_2 and b_3 are stored in *SRB2*. The multiplier by 3, implemented by adding and shifting, generates the data $3b_2$ and $3b_3$, 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 and Y_2 in the adder, Y_4 and Y_6 in the subtracter. The output mixer orders the even coefficient sequence of the ICT $<Y_0, Y_2, Y_4, Y_6>$. In this processor, the adder and the subtracter have an efficiency of 100%.

The processor $\mathbf{J_{4o}}$ architecture, shown in Figure 6, has been conceived to implement the computing diagram of Figure 3. This processor has five shift registers, ten multiplexers 4:1, and five arithmetic units with pipeline structure, operating in parallel at frequency $f_s/2$. Multiplications by 2 and by 8 are implemented as wired shift operations. The adder/subtracter *AE5* and the subtracter *AE6*, generate first the intermediate data $<d_0, d_1; d_2, d_3>$ from the input sequence $<a_7, a_6, a_5, a_4>$, stored in the register *SRA2*; d_0 and d_2 are stored in the register *SRD1*, while d_1 and d_3 are stored in *SRD2*. Next, from the data stored in *SRD1* and *SRD2,* in *AE5* the intermediate data, e_0 and e_3, are generated and in *AE6* e_1 and e_2; e_0 and e_3 are stored in *SRD1*, and e_1 and e_2 in *SRD2*. The adder/subtracter *AE7* and *AE8* generate first the intermediate data $<f_0, f_1, f_2, f_3>$ from the input sequence $<a_7, a_6, a_5, a_4>$, stored in the register *SRA2*; f_0 and f_2 are stored in the register *SRF1*, while f_1 and f_3 are stored in *SRF2*. After that, from the data stored in *SRD1*, *SRD2, SRF1* and *SRF2,* the intermediate data g_0 and g_2, are generated in *AE7* and in *AE8* g_1 and g_3; g_0 and g_2 are stored in *SRF1*, and g_1 and g_3 in *SRF2*. The output adder *AE9* generates the odd coefficients of the ICT from the intermediate data $<e_0, e_1, e_2, e_3>$ and $<g_0, g_1, g_2, g_3>$, stored in the registers *SRD1*, *SRD2, SRF1* and *SRF2*. These coefficients are produced in natural order: $<Y_1, Y_3, Y_5, Y_7>$. All the arithmetic elements of processor $\mathbf{J_{4o}}$ have an efficiency of 100%.

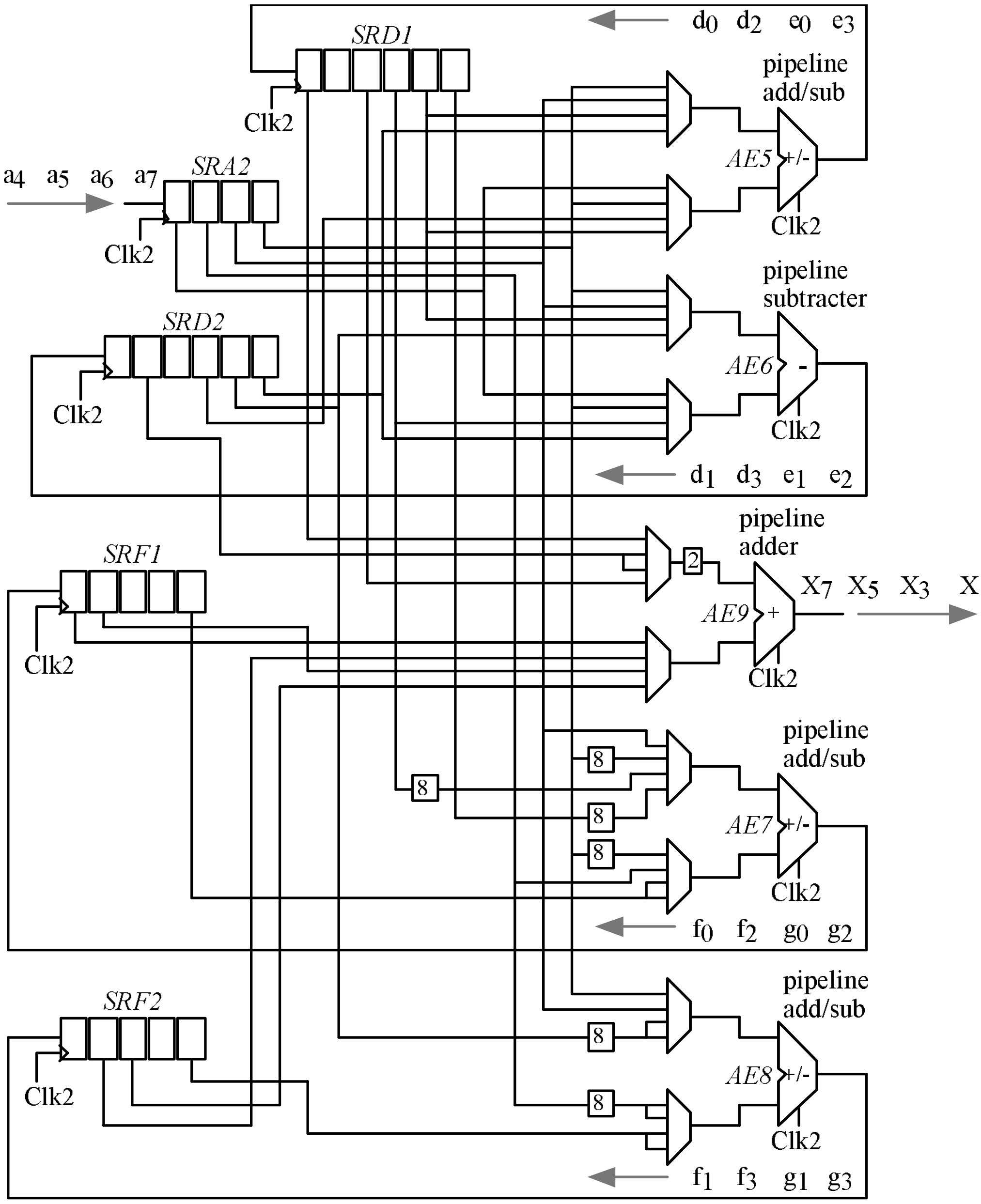

Figure 6. Architecture of processor $\mathbf{J_{4o}}$

4. 2-D ICT ASIC PROCESSOR

The 2-D ICT for a real input sequence **x** is defined as

$$\mathbf{X} = \mathbf{T}\ \mathbf{x}\ \mathbf{T}^t = \mathbf{K}\ \mathbf{J}\ \mathbf{x}\ \mathbf{J}^t\ \mathbf{K}^t = \mathbf{K}\ \mathbf{Y}\ \mathbf{K}^t \tag{16}$$

where **X** and **x** are 8x8 matrices. Reordering the input data and the transform coefficients applying the rules defined in (5) and (6) to both dimensions, the 2-D ICT can be expressed as

$$\mathbf{X}' = \mathbf{T}_R\ \mathbf{x}'\ \mathbf{T}_R{}^t = \mathbf{K}_R\ \mathbf{Y}'\ \mathbf{K}_R{}^t \tag{17}$$

the 2-D **J** transform being

$$\mathbf{Y}' = \begin{bmatrix} \mathbf{J}_{4e} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{4o} \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \mathbf{x}' \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \begin{bmatrix} \mathbf{J}^t_{4e} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{4o} \end{bmatrix} \tag{18}$$

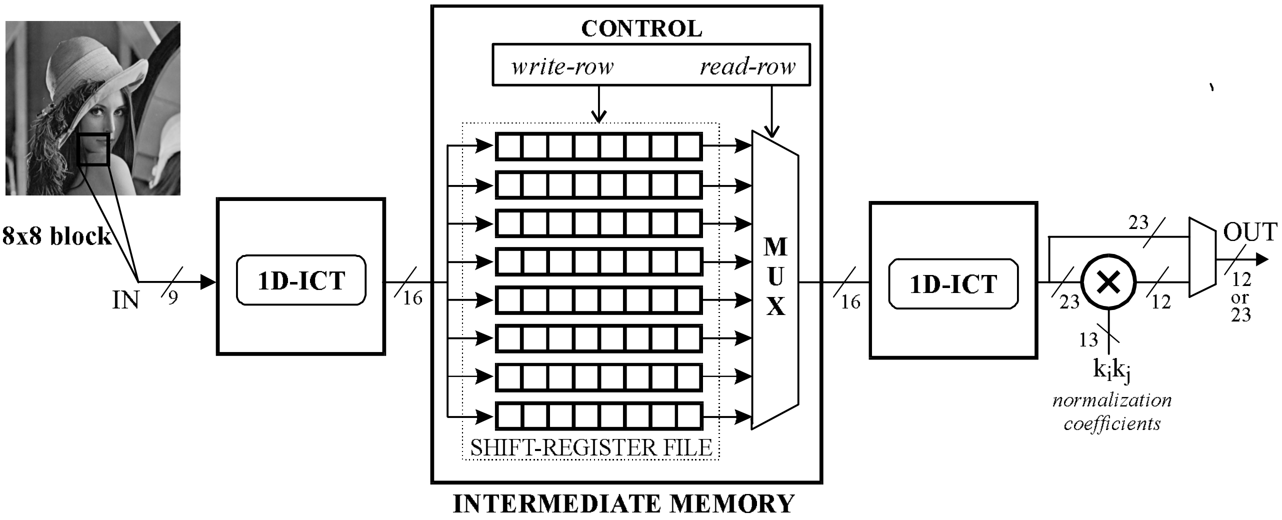

Figure 7. Scheme of 8x8 2-D ICT image processing

The 2-D ICT architecture shown in Figure 7 is based on a computing scheme derived from eq. (17) and (18), which first obtains the 2-D **J** transform followed by the normalization. This 2-D **J** transform is implemented using two 1-D **J**(10, 9, 6, 2, 3, 1) processors that share a memory to store the intermediate data. This memory, whose scheme shown in Figure 7, is made up of a shift-register file with 8x8 elements and a control circuit having a 7-bit pipeline counter and some simple additional logic. Reading and writing the shift register file is a two steps process: by row and by columns. In the first step, the output data of the first processor, corresponding to the eight 1-D ICT of an image block, are stored by rows, following a decreasing order, until the memory is filled up; at the same time, the second processor reads by rows the intermediate data stored in the register file, corresponding to the eight 1-D ICT of the previous image block. In the second step, columns store the intermediate data corresponding to the next image block, while the second processor reads by columns the intermediate data of the previous block.

Using the architecture shown in Figure 7, a 2-D ICT processor chip for transform image compression has been designed, being compatible with standard IEEE 1180-1990. The output coefficients can be selected without normalization (23 bits output) or with normalization (12 bits output according to standard IEEE 1180-1990). The design has been made with a semi-custom methodology, using the 0.35µm CMOS CSD 3M/2P 3.3V technology of *Austria-Microsystem* [14]. The circuit, whose microphotograph is shown in Figure 8, has an area of 2900x3200≈9.3 mm^2 (the core is 2200x2480≈5.5mm^2) with 13 input *pads*, 26 output pads and 22 power pads. The circuit contains 12446 cells, 6757 of them being flip-flops. The full functionality of prototype has been tested using a Logic Analysis System and has been verified at 200 MHz using a simplified test arrangement. From detailed circuit simulation, including line delays, a maximum operating frequency of about 300 MHz has been established.

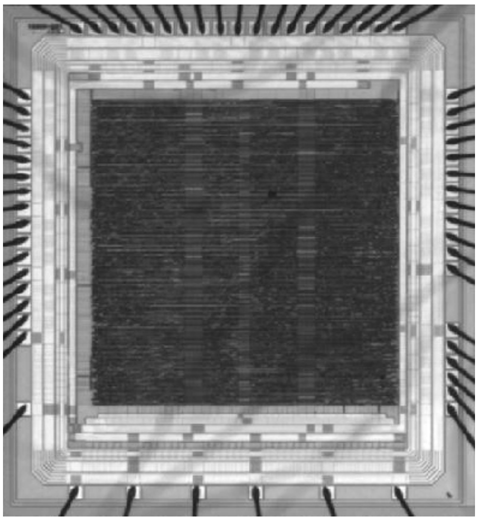

Figure 8. Microphotograph of 2-D ICT chip

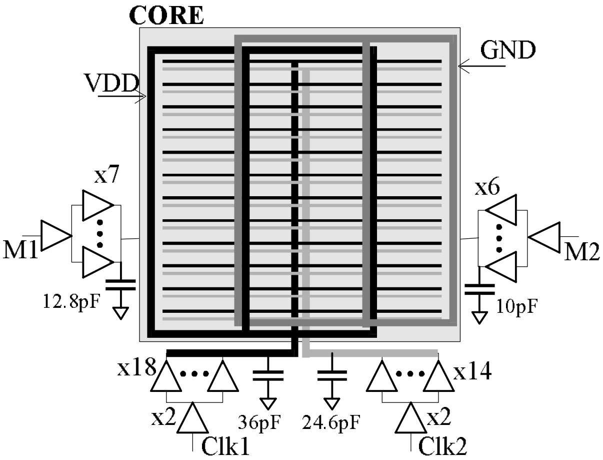

Figure 9. Distribution of control signals and main power rails

For adder/subtracter implementation, the binary look-ahead carry (BCL) adder [15] has been selected. The BCL adder keeps the number of gates along the critical path to $\log_2 n+2$ for an n-bit addition. Moreover, its structure is highly suitable for pipelining because it presents the same number of gates from each input to each output. All adders/subtracters in the processor are based on the BCL structure with one stage of pipelined registers in the middle to attain a high-speed addition. A highly pipelined output multiplier, having 3006 cells, performs the final normalization. The multiplier and the shift-register file account for nearly 50% of the total surface.

The processor has four main control signals: Clk1, external clock at frequency f_s, Clk2, internal clock at frequency $f_s/2$, and the multiplexer selection signals M1 at frequency $f_s/4$ and M2 at frequency $f_s/8$. Figure 9 shows the fan out of each of these signals and the scheme used to distribute them on the circuit. As this is a core-limited circuit, periphery buffers of 2mA drive strength have been placed in the pad ring. These buffers are short to create a single net and are placed side-by-side with their own power and ground supplies. To avoid clock skew, Clk1 and Clk2 are routed as a main trunk that goes through the middle of the chip, with branches on each side. M1 and M2 are globally routed in the chip. Power pads have been placed on all four sides of the chip to obtain an even current distribution to I/O pads, buffers and core. Internal buses and power feed thru cells are used to get a better current distribution to the standard cell area. The main supply buses of VDD and GND are also shown in Figure 9.

5. CONCLUSIONS

This paper presents an 8x8 2-D ICT(10, 9, 6, 2, 3,1) processor chip for image compression compatible with standard IEEE 1180-1990. This processor uses a minimal internal memory and a parallel-pipelined architecture to attain the maximum frequency of operation allowed by the technology estimated in 300MHz. As result, the architecture presents a low latency, high throughput and continuous data flow with a 100% efficiency in all computational elements. A prototype has been implemented in a 0.35-µm CMOS technology with an area of 9.3 mm^2. To avoid clock skew, clocks are routed using a clock-trunk strategy and the buffers are placed on the periphery. The prototype has been tested satisfactory using a test environment based on a Logic Analysis System. Simple test has been passed at 200 MHz.

ACKNOWLEDGMENT

We wish to acknowledge the Spanish Ministry of Science and Technology for the financial help TIC2000-1289 received to support this work.

REFERENCES

1. Chin-Liang-Wang, Chang-Yu-Chen, "High-throughput VLSI architectures for the 1-D and 2-D discrete cosine transforms," IEEE Trans. on Circuits and Systems for Video Technology, vol. 5, no. 1, pp. 31-40, February 1995.
2. W. Ku, "A cost-effective architecture for 8x8 two-dimensional DCT/IDCT using direct method," IEEE Trans. on Circuits and Systems for Video Technology, vol. 7, no. 3, pp. 459-67, June 1997.
3. Guoan-Bi, Gang-Li, Kai-Kuang-Ma, Tan-TC, "On the computation of two-dimensional DCT," IEEE Trans. on Signal Processing. vol. 48, no. 4, pp. 1171-1183, April 2000.
4. Hao-Chieh-Chang, Jiun-Ying-Jiu, Li-Lin-Chen, Liang-Gee-Chen, "A low power 8x8 direct 2-D DCT chip design," Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology. vol. 26, no. 3, pp. 319-332, Nov. 2000.
5. Ching-Hsien-Chang, Chin-Liang-Wang, Yu-Tai-Chang, "Efficient VLSI architectures for fast computation of the discrete Fourier transform and its inverse," IEEE Trans. on Signal Processing. vol. 48, no. 11, pp. 3206-3216, Nov. 2000.
6. Shen-Fu-Hsiao, Jian-Ming-Tseng, "Parallel, pipelined and folded architectures for computation of 1-D and 2-D DCT in image and video codec," Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology, vol. 28, no. 3, pp. 205-220, July 2001.
7. N. Merha, B. Vasudev, "A multiplication-free approximate algorithm for the inverse discrete cosine transform," Proc. 1999 Int. Conference on Image Processing, IEEE, vol. 2, pp .759-63, 1999.
8. Jie-Liang, T.D. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," IEEE Trans. on Signal Processing, vol. 49, no. 12, pp. 3032-44, Dec. 2001.
9. W.K. Cham,"Development of integer cosine transforms by the principle of dyadic symmetry," IEE Proc. Part I, vol. 136, no.4, pp.276-282, August 1989.
10. W.K. Cham, Y.T. Chan, "An order-16 integer cosine transform," IEEE Trans. on Signal Proc., vol. 39, no. 5, pp. 1205-1208, May 1991.
11. K.T. Lo, W.K. Cham, "New orthogonal transform for image compresión," Proc. 1993 IEEE Int. Symp. on Circuits and Systems, IEEE, vol. 1 , pp. 263-6, 1993.
12. Wai-Kuen-Cham, Chiu-Sing-Choy, Wai-Kit-Lam, "A 2-D integer cosine transform chip set and its applications." IEEE Trans. on Consumer Electronics, vol. 38, no. 2, pp. 43-47, May 1992.
13. Tin-Chak-Johnson-Pang, Chiu-Sing-Oliver-Choy, Cheong-Fat-Chan, Wai-Kuen-Cham, "A self-timed ICT chip for image coding," IEEE Trans. on Circuits and Systems for Video Technology, vol. 9, no. 6, pp. 856-860, Sept. 1999.
14. http://asic.austriamicrosystems.com
15. R.P. Brent and H.T. Kung, "A regular layout for parallel adders," IEEE Trans. on Computers, vol. C-31, no. 3, pp. 260-264, March 1982.