A 2D-DCT LOW-POWER ARCHITECTURE FOR H.261 CODERS

E. Scopa, A. Leone, R. Guerrieri and G. Baccarani

DEIS, Università di Bologna
Via Risorgimento, 2, I-40136 - Bologna
ITALY

ABSTRACT

A low-power architecture for 2D-DCT is presented. It has been designed for portable H.261-compliant videotelephone applications, but most of the results and considerations apply to MPEG systems too. The presence of quantization in the coding process has been exploited, adapting the precision of DCT calculations to the quantization noise level. The proposed architecture has the capability of dynamically controlling power consumption by reducing the precision to the minimum required level and turning off sub-systems when they are not necessary for the computation. Compared with a standard implementation, power consumption is reduced by a factor between 7 and 10, without appreciable degradation of the transmission quality.

INTRODUCTION

The two-dimensional Discrete Cosine Transform (2D-DCT) is used in video sequence coders, such as those based on the MPEG [1] or the H.261 standards [2], in order to remove spatial redundancy.

Many special-purpose ICs architectures have been developed to compute the DCT transform at high speed and high precision [3, 4, 5, 6], without significant constraints on power consumption. However, in portable telephone systems, reduction of power consumption is highly desirable. This is obtained in the proposed system using two different methods. The first is based on an algorithm that eliminates most of the redundant calculations, without any precision loss. This redundancy is a consequence of the differential coding of successive frames, which produces input values for the DCT transform that are usually very small or null. The second is based on the dynamic control of the accuracy of the computation. Since the 2D-DCT output values are successively quantized by a variable coefficient, computing the transformed coefficients with high precision is unnecessary in most cases. The H.261 Recommendation does not dictate any precision constraints on the 2D-DCT, whereas it requires very high precision for the inverse transformation, due to equipment compatibility problems. The proposed architecture is characterized by a precision-dependent power consumption that decreases when the quantization coefficient is larger.

The new technique uses ***Distributed Arithmetic*** (DA) [3] in conjunction with a fast ***Direct Bidimensional Approach*** (DBA). Furthermore, the intensive use of pipelining techniques permits the calculation of an 8×8 transformation every 13 clocks cycles: coding QCIF gray-scale images at 30 frames/s requires a clock speed of 154 KHz.

MATHEMATICAL STATEMENT

The 8×8 2D-DCT can be defined in the ***separable form*** as a product between matrices:

$$\underline{\mathbf{Z}} = \underline{\mathbf{C}}^t \, \underline{\mathbf{X}} \, \underline{\mathbf{C}} \tag{1}$$

where $\underline{\mathbf{C}}$ is an 8×8 matrix of constant elements, $\underline{\mathbf{Z}}$ and $\underline{\mathbf{X}}$ are the output and the input matrices, respectively. A fast version of the transformation involves a constant matrix $\underline{\mathbf{C}}_f$, with only 24 non-zero elements, at the cost of input pre-processing and output reordering:

$$\underline{\mathbf{Z}} = \underline{\mathbf{C}}^t \underline{\mathbf{X}} \, \underline{\mathbf{C}} = \underline{\mathbf{C}}^t (\underline{\mathbf{C}}^t \underline{\mathbf{X}}^t)^t = (\underline{\mathbf{R}} \, \underline{\mathbf{C}}_f \underline{\mathbf{S}})(\underline{\mathbf{R}} \, \underline{\mathbf{C}}_f \underline{\mathbf{S}} \, \underline{\mathbf{X}}^t)^t \tag{2}$$

where $\underline{\mathbf{S}}$ and $\underline{\mathbf{R}}$ are the pre-processing and reordering 8×8 matrices. Using the ***Kronecker Product*** $\otimes$ and the ***lexicographic transposition*** $L()$ and considering the relationship $L(\underline{\mathbf{A}}\underline{\mathbf{H}}\underline{\mathbf{A}}^t) = (\underline{\mathbf{A}} \otimes \underline{\mathbf{A}})L(\underline{\mathbf{H}})$, it is possible to write the transform in its direct bidimensional form

$$\underline{\mathbf{Z}} = L^{-1}(\underline{\mathbf{R}}_F \underline{\mathbf{K}}_F \underline{\mathbf{S}}_F L(\underline{\mathbf{X}})) \tag{3}$$

where $\underline{\mathbf{R}}_F = \underline{\mathbf{R}} \otimes \underline{\mathbf{R}}$, $\underline{\mathbf{S}}_F = \underline{\mathbf{S}} \otimes \underline{\mathbf{S}}$ and $\underline{\mathbf{K}}_F = \underline{\mathbf{C}}_f \otimes \underline{\mathbf{C}}_f$ ($\underline{\mathbf{R}}_F, \underline{\mathbf{S}}_F$ and $\underline{\mathbf{K}}_F$ are 64×64). The Kronecker product $\underline{\mathbf{H}} = \underline{\mathbf{A}} \otimes \underline{\mathbf{B}}$ defines a 64×64 matrix from two 8×8 matrices, with $H_{8u+x,8v+y} = A_{u,v} B_{x,y}$ where $u, v, x, y \in [0:7]$. The lexicographic transposition rearranges a matrix $\underline{\mathbf{M}}$ into a vector $\mathbf{V}$, so that, for 8×8

This work has been partially supported by the Italian Ministry of University and by SGS-THOMSON Microelectronics under the *National Program of Research on Bioelectronics Technology*.

0-7803-2431-5/95 $4.00 © 1995 IEEE

matrices it results $V_{8x+y} = M_{x,y}$, where $x, y \in [0:7]$. The block-diagonal matrix $\underline{\mathbf{K}}_F$ has 576 over 4096 non-zero elements. The DBA formulation (3) has been used because it allows a simple and efficient precision control. Only one step of calculation is required to evaluate the 2D-DCT, while the ***separable form*** requires two steps and one memory access.

THE ALGORITHMIC SOLUTION

The multiplication by matrices $\underline{\mathbf{R}}_F$ and $\underline{\mathbf{S}}_F$ is obtained by simple permutations, sums and subtractions. Therefore the attention can be focused on the product $\mathbf{Y}$ between a vector $\mathbf{X}$ of 64 variable elements and a constant sparse block-diagonal 64×64 matrix $\underline{\mathbf{K}}_F$.

Distributed Arithmetic

The DA allows us to implement the product without the use of multipliers: only ROM-access, add and shift operations are required. The DA implementation of the DBA 2D-DCT is given by

$$\mathbf{Y} = -2^{n-1}(\underline{\mathbf{K}}_F\mathbf{X}^{(n-1)}) + \sum_{i=0}^{n-2} 2^i(\underline{\mathbf{K}}_F\mathbf{X}^{(i)}) \quad (4)$$

where n is the binary width of the inputs and $\mathbf{X}^{(i)}$ is a vector of bits that contains the i^{th} bit (from LSB) of the element j of $\mathbf{X}$ in the j^{th} position.

Since the term $\underline{\mathbf{K}}_F\mathbf{X}^{(i)}$ is a product between a constant matrix and a vector of 64 bits, the possible results (2^{64}) could be stored in a ROM and addressed by the vector $\mathbf{X}^{(i)}$. Let $B_k^{(i)}$ be the k^{th} element of $\mathbf{X}^{(i)}$ and $\mathbf{K}_j$ the j^{th} row of $\underline{\mathbf{K}}_F$. If we define 64 ROMs, that contain the values

$$\mathrm{F}_j(\mathbf{X}^{(i)}) = \mathbf{K}_j\mathbf{X}^{(i)} = \sum_{k=0}^{63} K_{jk}B_k^{(i)}, \quad j = 0, ..., 63, \quad (5)$$

at the address $\mathbf{X}^{(i)} = \{B_{63}, B_{62}, ..., B_0\}$, the product $\mathbf{Y}$ can be expressed as

$$\mathbf{Y} = -2^{n-1}\mathbf{F}(\mathbf{X}^{(n-1)}) + \sum_{i=0}^{n-2} 2^i\mathbf{F}(\mathbf{X}^{(i)}), \qquad (6)$$

where $\mathbf{F}$ is the vector defined by (5).

Indeed this is simply an explanation of the DA technique. The proposed architecture does not require such widely-addressed ROMs, since all the needed data are stored in 144 4-bit addressable ROMs.

Redundancy Elimination

A number in 2's complement has always an initial string of one or more '0' if positive and one or more '1' if negative: the length of these strings increases when the absolute value of the number decreases. We can represent the same number deleting all the bits but one of this initial string. Beginning the DA application from the vector of the most significant bits $\mathbf{X}^{(n-1)}$, we skip all the steps until the one in which the vector $\mathbf{X}^{(p)} \neq \mathbf{X}^{(p+1)}$, e.g. when any of the inputs has a transition from 0 to 1 or 1 to 0 (Figure 1). This technique reduces significantly the real computational cost, because, as stated before, many DCT inputs are very small or null. Obviously the ROMs and the accumulators, each composed by pairs of adder/subtractor and register, must have a stand-by option with no power consumption.

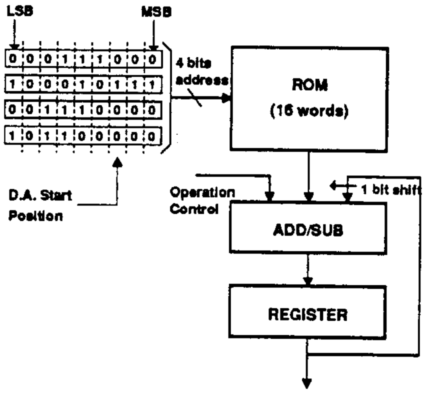


Figure 1: 4-bit DA Multiplier.

Precision Control

DA allows an efficient control of the precision through two parameters: the width w of the words read from the ROMs (***width control***) and the binary weight t of the last read word (***truncation control***). This means that the least-significant $(t-1)$ words are not read.

If the power consumption of the ROMs is proportional to the output width (i.e. ROM with bit-matrix partially pre-chargeable), the ***width control*** can further reduce the power consumption of the entire system. A similar technique, without dynamic control, has been used in [7] in order to reduce the hardware complexity of a 2D-IDCT IC.

The precision-control algorithm takes the quantization coefficient q as input: higher quantization parameters reduce the quality of the signal and, therefore, lower w and higher t can be used without further significant degradation. Simple rules $q \rightarrow (w, t)$, that guarantee no perceptible degradation in transmission quality, have been experimentally defined.

Therefore the controlled-precision non-redundant DCT is performed by the following algorithm

$$p = p(\mathbf{X}); \tag{7}$$

$$w = w(q); \quad t = t(q); \tag{8}$$

$$\mathbf{Y} = -2^{p-1}\mathbf{F}_w(\mathbf{X}^{(p-1)}) + \sum_{i=t}^{p-2} 2^i \mathbf{F}_w(\mathbf{X}^{(i)}), \tag{9}$$

where the elements of the vector $\mathbf{F}_w$ are represented by the w MSBs of the elements of $\mathbf{F}$.

Although the precision-control technique introduces noise in the 2D-DCT result, no unstable behavior of the H.261 coder/decoder loop has been found using this implementation.

THE ARCHITECTURE

Figure 2 shows a simplified structure of the proposed architecture. The Control Unit implements the timing control (not depicted in the figure) and the dynamic precision control. The Input Unit slices the 64 9-bit image coefficients into 9 64-bit words. In the 2D Pipelined Butterfly Unit, that implements the function of $\underline{\mathbf{S}}_F$, the sliced inputs are processed through different pipelines: more precisely, through one two-stage, two three-stage and one four-stage pipeline of 1-bit adders. The outputs of this unit are subdivided in four ***channels*** (A, B1, B2 and C), grouping the lines with the same delays. This grouping allows a more efficient reduction of redundant calculations since the lines of the same group have the same ***transition*** probability.

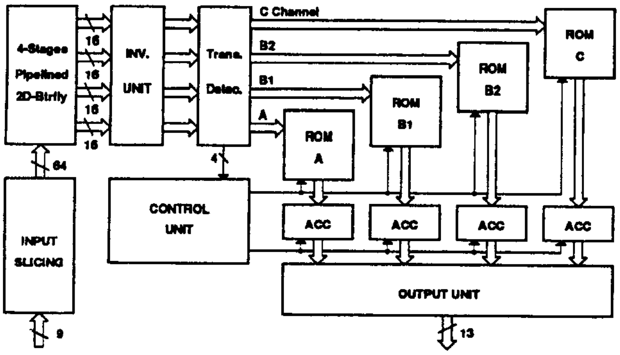


Figure 2: Architectural scheme.

Since the DA application must begin from the MSB while the previous unit sends out the bits starting from the LSB, the Inversion Unit reverses the order of the bit vectors, without affecting the relative delays between the channels. This stage is double-buffered: using two sets of 64 13-bit bidirectional shift registers, we do not need to stop the data-flow, since one set loads the data from the 2D Pipelined Butterfly while the other one sends out the 13 64-bit words of the previous transform. Their roles are exchanged after each transformation.

The Transition Detectors signal to the Control Unit the first transition in any of the four channels starting the DA application on this channel.

In the ROMs the necessary 144 × 16 pre-calculated 9-bit sums of the elements of $\underline{\mathbf{K}}_F$ are stored. The ROMs are subdivided in 4 groups, one for each channel. In the groups **B1**,**B2** and **C** the ***partial sum*** technique [3] has been used to reduce the size of the ROMs and the number of necessary words down to 2304. The partial sum technique is implemented by a pipeline of adders (one-stage for groups **B** and two-stage for group **C**), not shown in figure. All the channels have the same delay after this stage.

The DA accumulation unit reconstructs, for each channel, the final transform result from the partial ones extracted from the ROMs. In order to maintain a high throughput efficiency, a set of registers loads the final results from the accumulators, so that they are available for the next transform. The Output Unit reads these registers, which are connected via 3-state buffers to the same bus, and sends out the transformed coefficients in the desired zig-zag order.

The Input and Output units work at the pixel rate, while all the other units work at about $1/5^{th}$ of this speed. The architecture is fully pipelined: if the pixel differences are represented by 9 bits, every 13 internal clock cycles, one 8 × 8 block is processed.

SIMULATION RESULTS

The proposed architecture has been carefully simulated within a software implementation of an H.261 coder. We can assume that power consumption grows up with the average number of bits extracted from the ROMs. An architecture based on the classical 2D-DCT separable form, such as the one described in [3] (scaled down to the 8 × 8 2D-DCT with 9-bit ROM word), reads 13824 bits per transformation. The new architecture performs the calculation reading less than 1800 bits. Moreover it can be noticed that also the accumulators can be turned off when the ROMs are not accessed. The calculation redundancy elimination and the truncation control reduce the average number of DA steps from the worst case of 13, down to 2.5-3.5. The remaining power consumption reduction is obtained by width control. Fig. 3 shows the power consumption reduction averaged over the whole 150-frame sequence of Miss America as a function of the quantization factor. It can be seen how reduction in power consumption increases from a minimum of 3.7, when the quantization factor is 1, up to 17.3 for the highest quantization factors. The behaviour of this relationship derives from the formulation of expressions (8).

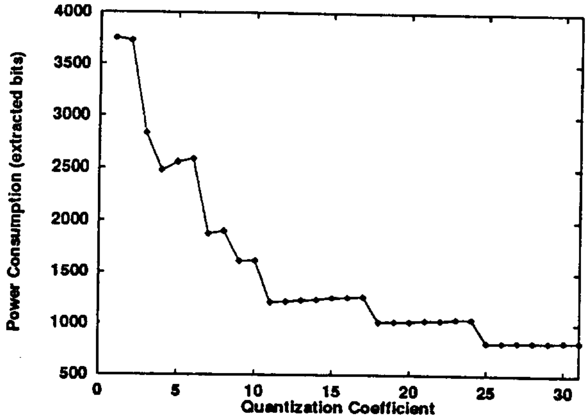
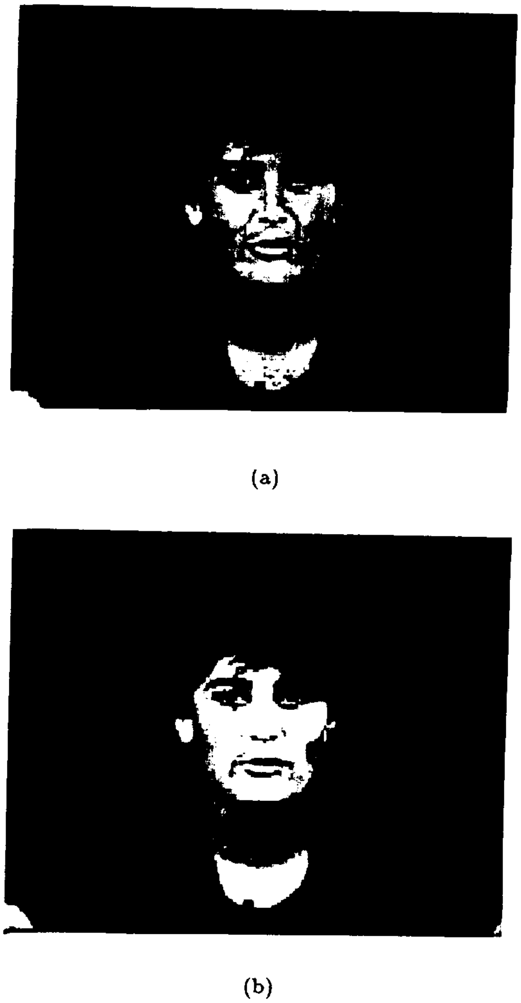


Figure 3: Power consumption comparison as a function of the quantization coefficient.

A sample frame of the test sequence coded by the full precision DCT is shown in fig. 4a, while the same frame coded by the low power DCT is shown in fig. 4b. No degradation can be noticed when the low power DCT architecture is used.

Fig. 5 shows the peak signal to noise ratio of a test sequence encoded by an H.261 coder using the proposed DCT and the full precision DCT: on the average only a fraction (0.05-0.2) of dB is lost.



(a)

(b)

Figure 4: Frame n. 33 of the sequence "Miss America" coded by the full precision (a) and low power DCT (b).

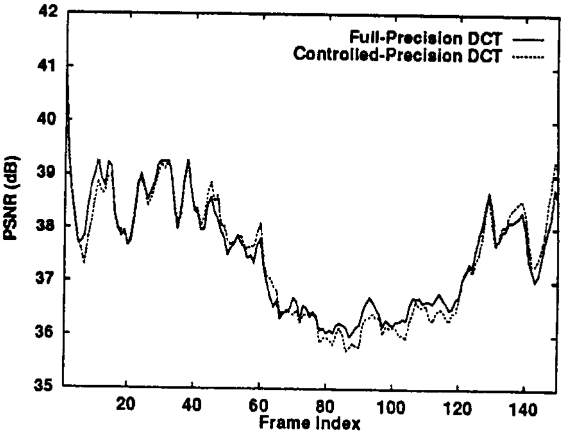


Figure 5: PSNR comparison between full precision and controlled precision DCT - Sequence "Miss America"

REFERENCES

[1] D. LeGall, "MPEG: A video compression standard for multimedia applications," *Comm. of ACM*, pp. 47–58, Apr. 1991.

[2] M. Liou, "Overview of the p*64 kbits/s video coding standard," *Comm. of ACM*, pp. 60–63, Apr. 1991.

[3] M. T. Sun, T. C. Chen and A. M. Gottlieb, "VLSI Implementation of a 16x16 Discrete Cosine Transform" *IEEE Trans. Circuits Syst.*, vol.36, pp.610-617, 1989

[4] S. Uramoto et al. "A 100-MHz 2-D Discrete Cosine Transform Core Processor" *IEEE J. Solid-State Circuits*, vol.27, pp.492-499, 1992

[5] P. A. Ruetz et al. "A high-performance full-motion video compression chip set," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 2, pp. 111–122, June 1992.

[6] H. Fujiwara et al., "An all-ASIC implementation of a low bit-rate video codec," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 2, pp. 123–134, June 1992.

[7] P. A. Ruetz and P. Tong, "A 160-Mpixel/s IDCT Processor for HDTV" *IEEE Micro*, pp.28-32, 1992