A MEMORY SYSTEM SUPPORTING THE EFFICIENT SIMD COMPUTATION OF THE TWO DIMENSIONAL DWT

Maria A. Trenas, Juan López and Emilio L. Zapata

Dept. Computer Architecture
University of Málaga. SPAIN
{maria,juan,ezapata}@ac.uma.es

Francisco Argüello

Dept. Electronics and Computation
University of Santiago. Spain

ABSTRACT

Real time image processing uses SIMD engines to accelerate the computation of algorithms as DCT, FFT or DWT. So, a good skewing scheme becomes essential for avoiding memory bank conflicts. In this paper a memory system is introduced for the efficient in-place computation of such transforms. It consists of $M = 2^m$ memory modules, providing parallel access to M image points whose patterns are a row or a column, the interval in both cases being 2^l, $l \geq 0$. The efficiency of our design is proved through the computation of the 2D-DWT.

1. INTRODUCTION

Image processing often uses SIMD engines to exploit the parallelism inherent to some typical operations. Video processor chips (i.e. VLSI chips for multimedia processing and transmission) are a clear example [6]. In this context, a memory system with high bandwidth becomes essential.

The in-place computation of two dimensional orthogonal transforms, of great interest in image processing, could be accelerated if the memory system allows parallel access to elements in a row, column or sub-matrix of the image, with interval 2^l. To achieve this goal, standard interleaved memory systems made of a number of banks power of two must be discarded because the required elements cause bank conflicts. Memory systems involving a prime number of memory modules have been proposed for accessing to a number of conflict-free patterns. Voorhis and Morrin [5], presented a system with pq o $pq + 1$ banks, allowing the access to blocks of $p \times q$ adjacent elements in a row, column or sub-matrix. Park [4] improved this design by simplifying the memory address generation circuit in the case of $pq + 1$ modules. Recently, new memory systems have been proposed to provide these access patterns. Park and Harper [2] have extended Park's design [4], using $2^m + 1$ banks memory system, allowing parallel access to 2^m points from a row with stride 2^l, whilst in the case of columns and submatrices the only permitted interval is 1.

But using a prime number ($2^m + 1$) of memory modules complicates the assignment (or skewing) functions required to calculate the bank and address corresponding to each element, due to the arithmetic involved, so the addressing and data alignment circuits become complex. Besides, it results in a waste of memory storage space. To avoid these drawbacks, Deb[3] has recently proposed a system with $M = 2^m$ banks, applying different assignment functions to different sections of the image matrix (multiskewing). This design permits the access to 2^m points with a large class of patterns (rows, columns, diagonals, coils, etc.) using simple circuits. However, he only considers the case in which the image matrix has dimension $N \times N$ and the number of banks is N.

In this work we introduce a memory system with M=2^m memory modules, so that it requires simple memory bank and address assignment functions. It allows conflict-free parallel access to M points in any row or column from an image of dimension NxN ($M = 2^m < N$), being 2^l, $l \geq 0$, the interval between two consecutive elements . Thus, the systems appears to be well fitted for the 'in-place' computation of two dimensional transforms like FFT, DCT, Gaussian Pyramid and, specially, wavelets in SIMD computers.

Section 2 briefly introduces the DWT. In section 3 the memory system is described and several accessing patterns are discussed. The computation of the DWT in a SIMD machine using the proposed memory system is studied in section 4. Finally, in section 5, we establish the main conclusions.

2. AN APPROACH TO DWT

Wavelets appear in a wide range of areas, specially in the context of computer graphics, due to its temporary and frequency characteristics [1]: image coding and compression [7][8], image processing using multiresolution techniques, modeling of curves and surfaces, radiosity computations, etc.

The unidimensional DWT splits the original signal into

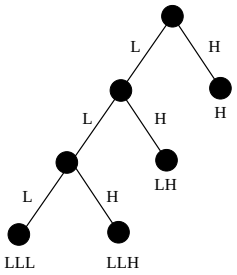

Figure 1: *Dyadic binary tree*

a high-pass and a low-pass signal. The splitting process goes on, in a recursive way, always being applied over the low-pass signal. This way, we obtain a pyramid with a maximum number of levels equal to $log2(N)$, see figure 1.

At each level of the pyramid, an input sequence $x(n)$ is filtered by using low-pass and high-pass filter coefficients $h(n)$ and $g(n)$ respectively:

$$y(n) = \sum_k h(k)x(2n+k) \qquad n,k \in Z \tag{1}$$

$$z(n) = \sum_k g(k)x(2n+k) \qquad n,k \in Z \tag{2}$$

The sequence $h(n)$ is the smoothing or scale filter, while the sequence $g(n)$ is the detail or wavelet filter.

The number of filter coefficients is very variable.In this paper DAUB4, a four-coefficient Daubechies wavelet [9] is used for simplicity. This won't suppose a loss of generality.

The computational structure of the in-place pyramidal algorithm is shown in figure 2. White circles represent the results of the high-pass filters (z_i) while black circles represent the original sequence at level 0, and the points $y(i)$ obtained after the low-pass filtering in further levels. These last points will be used when computing a new level l. The figure shows the increasing interval, of value 2^l, between them.

Most of the usefulness of DWT rests on the fact that it can severely truncate turning into sparse results [9]. Many negligible coefficients appear in the high-pass octaves obtained, reducing the amount of information to deal with, so that going on with the iteration becomes unnecessary in many applications. So, just a few first levels are usually computed, typically from 3 to 7.

The two dimensional DWT is attained by applying independently the low-pass and high-pass filters to both the rows and columns of the initial image. This way, we obtain four new images, depending on the pair of filters employed in its generation, four times smaller than the initial one. If we would permute the results after each filtering stage, grouping the results from the high-pass aside the results from the low-pass, we would arrive to a point distribution as the one

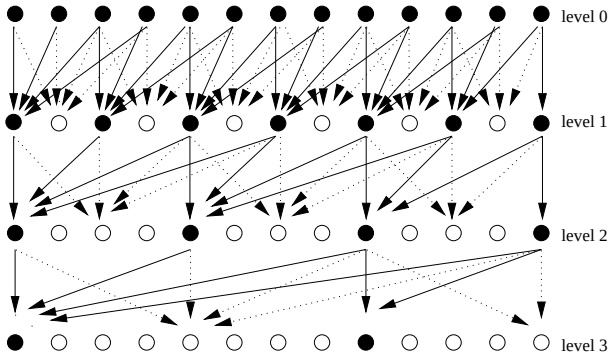

Figure 2: *Computational dependences in a four coefficients DWT.*

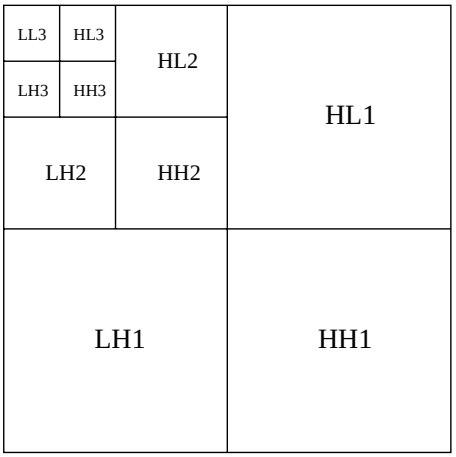

Figure 3: *The octaves of a three levels two dimensional DWT*

in figure 3. In this figure, HLk for example, is the result of a high-pass filtering of the rows, and a low-pass filtering of the columns, at level k of the pyramid.

However, working 'in-place', as we do, we will arrive to the situation depicted in figure 4. We show the original matrix (level 0) and the two first levels. Again, we represent as black circles the points that will take part in the following computations: those that result from the low-pass filtering of rows and columns. Again, there is an interval 2^l between the points to be accessed at level l, but in this case in both the rows and the columns.

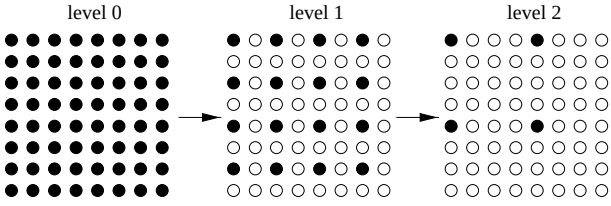

Figure 4: *Points to be accessed for levels* $l = 0..2$ *(* $N = 8$ *).*

3. THE MEMORY SYSTEM

A description of the memory system involves defining a *memory module assignment function* placing the image points to be simultaneously accessed in distinct memory modules, and an *address assignment function*, allocating an address inside the assigned module. In general, a system with a number power of two of memory modules, simplifies the hardware design of the addressing and routing circuits. Now we will describe the assignment functions we use.

The *memory module assignment function* assigns to the element at position (i, j) of the image matrix the following module number:

$$\mu(i,j) = [\sum_{h=0}^{k-1} (i_h + j_h)] mod M \tag{3}$$

i_h and j_h being the digits of the M-base representation of indices i and j, respectively. Obviously, $k = \lceil \frac{n}{m} \rceil$.

We point out the most important properties of the above distribution:

- This skewing scheme is not periodic, so it is not lineal.
- It can be considered as a multiskewing scheme. For index values $i, j < M$ it applies $\mu_0(i,j) = (i_0 + j_0) mod M = (i + j) mod M$. The image matrix is divided into blocks of size $M \times M$. The skewing scheme for the block at position (α, β) is defined as $\mu_{\alpha,\beta}(i,j) = [\mu_0(i_0, j_0) + \alpha + \beta] mod M$, where $\alpha = i_{k-1} + i_{k-2} + \ldots + i_1$ and $\beta = j_{k-1} + j_{k-2} + \ldots + j_1$. Local skewing are linear and periodic schemes.
- The distribution permits simultaneous access to M elements belonging to the same row or column of the image I(*,*) with any interval power of two, as long as the initial position (i, j) will be a valid one.

In other words, the horizontal (VH) and vertical (VV) accessing vectors allowed are:

$$\begin{aligned} VH_{2^l}(i,j) &= \{I(i, j + 2^l b) | 0 \le b < M\} \\ & 0 \le i < N, 0 \le j < N - 2^l(M-1), l \ge 0 \\ VV_{2^l}(i,j) &= \{I(i + 2^l a, j) | 0 \le a < M\} \\ & 0 \le j < N, 0 \le i < N - 2^l(M-1), l \ge 0 \end{aligned} \tag{4}$$

where 2^l is the stride to apply and auxiliary variables a and b are used for sweeping the accesses vector. The initial position (i, j) must verify the following:

$$\begin{aligned} HV_{2^l} &: \quad \tfrac{j}{2^l} mod M = 0 \\ VV_{2^l} &: \quad \tfrac{i}{2^l} mod M = 0 \end{aligned} \tag{5}$$

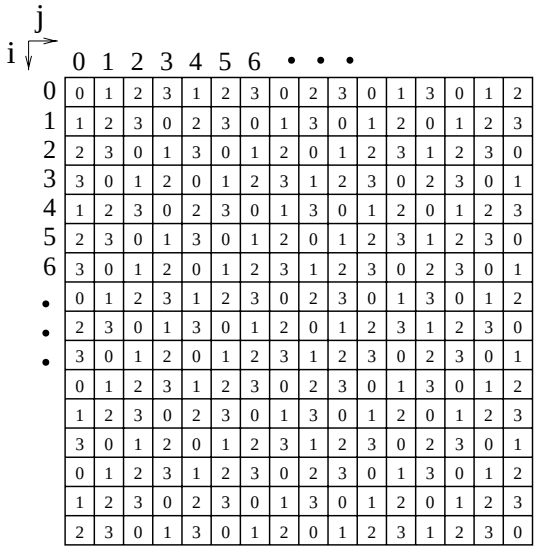

Figure 5: *Distribution matrix for* $N = 16$ *and* $M = 4$

The *address assignment function* is the usual for interleaved memory systems:

$$\alpha(i,j) = \lfloor s/M \rfloor \tag{6}$$

Equations (3)(6) guarantee that elements mapped into the same memory module will be assigned different addresses. Besides, it is assured that there won't be holes of unused memory space in the memory system. No demonstration of the above assertions is given for simplicity.

An example of the proposed skewing scheme is shown in figure 5, where the matrix dimension is $N = 16$, and the number of memory modules is $M = 4$. Observe the recursive data distribution.

Let's see that, as an example, we can access to elements in a column, with initial position $(0, 3)$ and stride equal to 2. The accessed elements would be $[I(0,3) I(2,3) I(4,3) I(6,3)]$ assigned to memory modules 3, 1, 0 y 2, respectively. Since the initial position verifies (5), it is a permitted one and the memory access is conflict-free.

4. DWT COMPUTATION USING THE PROPOSED MEMORY SYSTEM IN A SIMD ARRAY

In this section we sketch the computation of a two dimensional DWT using the proposed memory system in a standard SIMD array, with the following basic architecture: P independent processing elements (PEs), mastered by a central control unit, and connected to M independent memory modules by an interconnecting network. Besides, we will suppose $M = P$ in the following paragraphs.

Figure 6 depicts a part of the accesses diagram involved in the two first levels of DWT computation, for the case

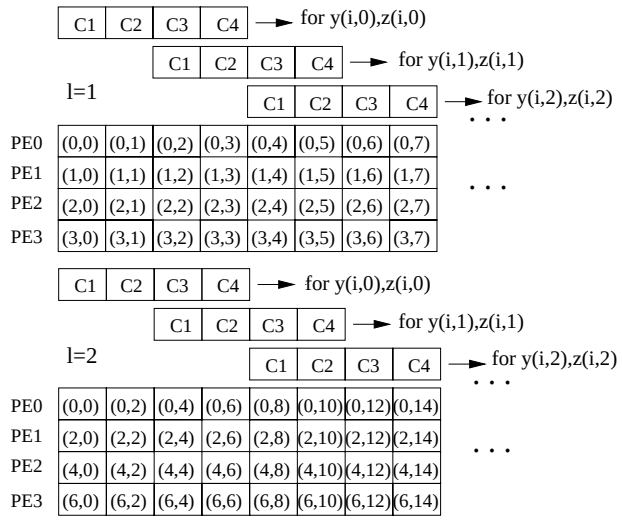

Figure 6: *Some accesses for row processing when building levels 1 and 2 ($M = 4$ and $N = 16$)*

$M = 4$. Only the filtering by rows is shown, as the one by columns can be obtained just by transposing the indices. A row is assigned to each PE that sweeps it, so that previously read values can be reused. Each PE applies coefficients C_i (representing sequences h(n) and g(n), for the low and high pass filtering respectively), to the four read values of $I(*,*)$. For example, PE0 at $l = 1$ reads points $[I(0,0)I(0,1)I(0,2)I(0,3)]$, and computes y(0,0) and z(0,0), that are stored in positions $I(0,0)$ and $I(0,1)$ of the array, respectively. Usually, no idle-processors will appear as it will be $N >> M$ and not all levels are computed.

As observed before, we can see in figure 4 how the 'in-place' computing of the DWT involves power of two strides both in the rows as in the columns: the distance between points to be accessed is multiplied by two with each new level computed. Computing 'in-place' will result in a different data distribution than in figure 3. However the elements belonging to each of the octaves can be easily localized.

5. CONCLUSIONS

SIMD engines are used for accelerating computation of many image processing algorithms. This can be achieved through the use of specific memory systems (or skewing schemes) permitting parallel access at each computation stage.

In this paper we propose a memory system that allows simultaneous access to elements in a row or a column, with whatever stride equal to a power of two. The hardware needed to implement the memory module and address functions is very simple because the number of memory modules is a power of two and the simplicity of our skewing scheme. The efficiency of our design is proved through the computation of the 2D-DWT.

The design is well suited for computing algorithms with a dyadic subband binary tree structure, as the DWT, and other two dimensional orthogonal transforms of great interest. Some of this transforms can be computed using a memory system just allowing parallel access to rows or columns with only stride 1 [4], or enabling the access by rows with any interval power of two as [2]. We have verified that our system reaches similar performance than [2] for Gaussian Pyramid, though using less memory modules and without producing holes that would waste the memory space available.

We must point out that the interconnection network shouldn't be a complex one (i.e. crossbar). Our actual work in this direction concludes that it may just consist of a rotation stage, and a stride-dependent number of perfect-shuffle ones.

6. REFERENCES

[1] Alain Fournier. Wavelets and their Applications in Computer Graphics. SIGGRAPH'95 Course Notes.

[2] J. W. Park, D. T. Harper. An Efficient Memory System for the SIMD Construction of a Gaussian Pyramid. IEEE Transactions on Parallel and Distributed Systems, vol.7, no. 8, August(1996),pp.855-860.

[3] Ashoke Deb. Multiskewing-A Novel Technique for Optimal Parallel Memory Acces. IEEE Transactions on Parallel and Distributed Systems, vol.7, no. 6, June(1996),pp.595-604.

[4] Jong Won Park. An Efficient Memory System for Image Processing. IEEE Transactions on Computers, vol. 35, no. 7, July(1986), pp.669-674.

[5] David C. Van Voorhis, Thomas H. Morrin. Memory Systems for Image Processing. IEEE Transactions on Computers, vol. 27, no. 2, February(1978), pp.113-125.

[6] Horng-Dar Lin. Handling Multimedia Information with VLSI. Circuits & Devices, July(1995), pp.25-31.

[7] Jerome M. Shapiro. Embedded Image Coding Using Zerotrees of Wavelet Coefficients. IEEE Transactions on Signal Processing, vol. 41, no. 12, December(1993), pp.1445-1462.

[8] Ricardo de Queiroz, C.K. Choi, Young Huh, K.R. Rao. Wavelet Transforms in a JPEG-Like Image Coder. IEEE Transactions on Circuits and Systems for Video Technology, vol.7, no.2, April(1997),pp.419-424.

[9] William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery. Numerical Recipes in Fortran. Second edition, Cambridge University Press.