A 500 MHz 2d-DWT VLSI Processor

A. Brizio, G. Masera, G. Piccinini, M. Ruo Roch, M. Zamboni*
Dipartimento di Elettronica, Politecnico di Torino
Corso Duca degli Abruzzi, 24 — I10129 Torino, Italy
Tel: +39 11 564 4004 — Fax: +39 11 564 4112
e-mail: brizio@polito.it

ABSTRACT

The Discrete Wavelet Transform (DWT) is a well-known mathematical tool, which has been proved to offer very good results for Digital Signal Processing applications. The DWT requires a massive computation, especially when real-time constraints are to be met, as it happens in a wide spectrum of applications; therefore dedicated processors are needed to achieve the necessary performance. In this paper a very high-speed VLSI implementation of a two-dimensional DWT processor suitable for image processing is presented.
The architecture is based on a bit-serial arithmetic; linear systolic arrays are employed for row filtering and parallel arrays for column filtering. Each basic building block of the arrays has been designed using a True Single Phase Clocking logic, then a manual place and route of the derived cells has been carried out. The processor works at 500 MHz and occupies less than 50 mm^2 in a 0.7 μm CMOS technology.

1 Introduction

The Discrete Wavelet Transform plays a major role in many image processing applications, however few VLSI implementations of the DWT have been described until now, in particular for the two-dimensional case. In section 2 a short review will be given of the wavelet theory, while known hardware implementations are briefly described in the following.

The architecture presented in [1] has been designed to employ the length-4 Daubechies' filter and cannot be easily adopted for other wavelet basis. In [2], Parhi and Nishitani developed the concept of folded architecture and proposed a digit-serial solution, both having a minimal number of registers.

More recently, in [3], several architectures have been presented: a SIMD solution, a systolic array and a parallel filter. The first architecture is very effective but his complexity is proportional to the input image size and the control circuitry has to be carefully designed. Both the systolic array and the parallel solution are attractive for VLSI implementation, because of their high regularity.

The systolic-parallel structure in [4] requires an internal memory and a converter that reorders the outputs of the row filters before passing them to the column filter. The second drawback can be overcome if all the outputs of the row filters are available at the same time, as suggested by Chuang and Chen in [5].

The idea is to use as many row filters as necessary to process the whole image at the same time and to allocate all the resources required to compute the different levels of the DWT, so that also the internal RAM can be dropped. This solution finds a practical limit in the size of typical images; this suggests the data partitioning in non overlapping groups of rows to be processed in a pipelined way in the processor: in figure 1 an N-rows image is decomposed in n blocks of q rows each. What arises from a feasibility analysis, however, is that the number of rows in each block has to be kept very small because of I/O limitations; this reduces the effectiveness of the solution and results in a less regular design.

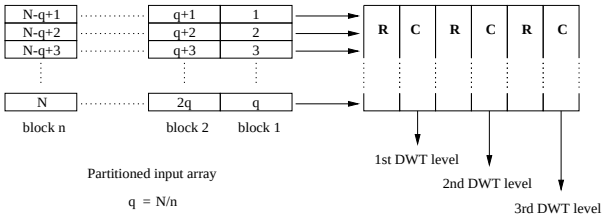

Figure 1: Partitioned scheme of the DWT processor: **R** are linear systolic arrays, **C** are parallel arrays

The strategy proposed in this paper to maintain an high degree of row parallelism is to adopt a serial data I/O and also a serial data processing, with a deep pipelining of the computation. Both these features well meet the design style recommended when using True Single Phase Clocking logic, allowing to keep a low level of fan-out and local interconnections.

In the following, after a brief review of the Wavelet Transform, section 3 presents the architecture of the

*This work was partially supported by the VLSI ARCHITECTURES Research Program of MURST Italy and by CSTV (TV Study Centre of the CNR)

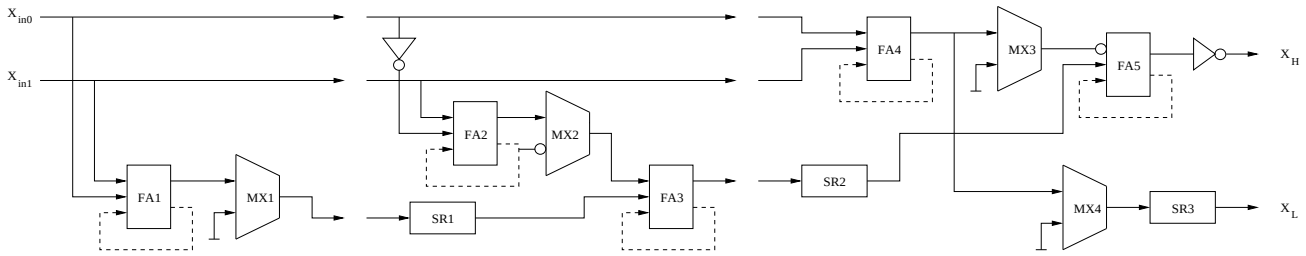

Figure 2: Cells of the linear systolic array

cells which compose the linear and parallel arrays for rows and columns filtering, while in section 4 the VLSI implementation is detailed. Final considerations and perspectives for future works are provided in the conclusions.

2 Discrete Wavelet Transform

From an algorithmic point of view the DWT is computed by an iterative filtering process [6]. At each iteration, or level of the transform, the input samples are filtered by a pair of high-pass and low-pass filters and then subsampled by a factor of 2, so that the total number of samples remains the same. The results of the so-called "analysis" process are a coarse reproduction of the original stream (the low-pass filtered version), which is used as the input sequence for the subsequent level, and an additional detail (the high-pass filtered data), which represents the DWT of the original sequence at the current level.

If the filters are designed to meet a particular set of constraints, the process is reversible and the original sequence can be reconstructed by filtering the data computed in the analysis process through a pair of high- and low-pass filters (possibly different from the previous one). This is referred to as "synthesis": starting from the coarsest approximation available and the corresponding level of detail, a better approximation at half the scale is obtained; the process is iterated until the original sequence is reproduced. At each level the inputs have to be upsampled by a factor of 2, by insertion of 0's in the sequences, before actual filtering.

In the two-dimensional case, the filtering is almost always performed by separate processes along the rows and the columns of the input array. This reduces the computational complexity with respect to the non-separable filtering; moreover, two-dimensional filters do not achieve better performance than mono-dimensional ones. Among the possible filter pairs, it has been shown that biorthogonal filters are often preferable to the orthonormal ones [7], especially for image compression applications. In fact, because of their liner phase characteristic, biorthogonal filters usually produce a better low resolution approximation of the original image, allowing a graceful degradation of the quality.

The filters' length is another parameter to be considered before to proceed with the design. While the orthonormal low-pass and high-pass filters have the same length, this is not true for the biorthogonal; this fact can have an impact on the architectural complexity, leading to a less regular design. In addition, very short filters are to be preferred from the hardware point of view, but their performance are often very poor.

In the proposed architecture, the low complexity has been privileged and a simple yet efficient biorthogonal filter, the so-called Two-Six, has been employed. Its coefficients are

$$h = \left[+\frac{1}{2}, +\frac{1}{2}\right]$$

$$g = \left[+\frac{1}{16}, +\frac{1}{16}, -\frac{1}{2}, +\frac{1}{2}, -\frac{1}{16}, -\frac{1}{16}\right]$$

where h indicates the low-pass analysis filter and g the high-pass one. This filter has an interesting feature: their coefficients are integer powers of 2 so that actual multipliers can be avoided in the VLSI implementation.

3 Processor architecture

For the chosen architecture, the number of required I/O pins can be expressed by the relation $p = (2q+2q) \cdot k/m$, where q is the number of rows in each processed block, k is the number of bits per pixel and m is the data I/O parallelism, so that m out of the k bits of a pixel feed the processor at the same time.

It is easily seen that a physical realization is impractical when m is equal to 1 (fully data parallelism), even for small q. Moreover, if q is too small, many partial sums are produced and need to be stored until the next group of rows is processed. The only possible alternative is to increase m, thus reducing the data parallelism and the throughput by a factor $1/m$. To the same extent, it is worth noting that True Single Phase Clocking logic (TSPC) [9] circuits tends to occupy a bigger area with respect to traditional static structures.

These considerations justify the choice $m = k$, that is a fully serial approach for data I/O, but also a bit-serial computation, which results in a better trade-off between chip area and pin-count. The proposed solution allows to have a rather high value for q, each block is composed by 32 rows, with 128 pins allocated for input and output of data. Given the high clock frequency, the throughput

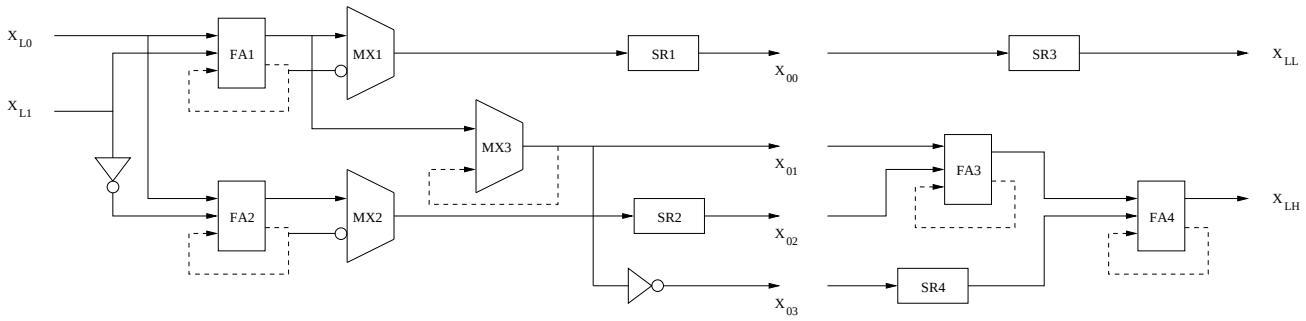

Figure 3: Cells of the parallel array

reduction is a minor disadvantage and the serial data processing allows to avoid the use of circuitry dedicated to the serial-to-parallel and parallel-to-serial data conversion. The time required to compute two levels of DWT for a $N \times N$ image is given by

$$\frac{N}{2} \cdot \frac{N}{32} \cdot \left(1 + \frac{1}{4}\right) \cdot T_{CK} \cdot k \simeq \frac{N^2 \cdot T_{CK}}{50}\, k$$

where k is equal to 12 and T_{CK} is 2 ns @ 500 MHz.

The linear array for rows filtering is composed of three cascaded cells, while two are the parallel array's basic stages. The building blocks are shown in figure 2 for the linear array and 3 for the parallel one: the filtering operation only requires full-adders (FA) and multiplexers (MX), due to the particular coefficients of the employed filter; the shift registers (SR) are delay lines that synchronize the bit flow among different paths.

The entire linear systolic array has been redesigned with respect to a previous implementation [8], because the number of bit per coefficient of the transform has been raised to 12 in order to improve the precision: three bits are devoted to the decimal representation. To reduce the total number of delay elements, the input data flow has been reversed: for each row, the two rightmost pixels enter in the processor first, and the shift registers have been moved from the input lines to the outputs of the high pass filter stages, so that the proper data synchronism is obtained delaying the partial sums instead of the input bits. This solution employs a total number of 27 flip-flops against 60 for each linear array, and allows to save about 4 mm^2 in the final circuit. An additional advantage is the reduced capacitive load on the clock lines, which allows to shrink the area requested by the clock buffering stages. Moreover the latency has been decreased and the control circuitry has been simplified, because the multiplexers MX1 and MX3 operate synchronously and also MX2 and MX4 can be controlled by the same signal.

As sketched in fig. 2, two homologous bits belonging to two successive pixels of an image row feed at each clock cycle the first cell of the linear array, which computes two convolutional products for the high-pass filter. The full adder FA1 computes the sum in a serial fashion starting with the least significant bits, while the multiplexer MX1 implements the division by 16 by inserting four 0's as the most significant bits. At the same time the second cell computes the products required by the two central coefficients of the high-pass filter, which are then added to the output of the previous cell, delayed by SR1. The final cell looks much like the second one, but it also has a lower branch in which the low-pass output is computed. The coefficients x_L and x_H are synchronized by the length 4 shift register SR3.

In the parallel array, the first cell receives a pair of low-pass (or high-pass) coefficients from the linear array (x_{L0} and x_{L1} in the picture) and produces four partial sums (x_{00}, x_{01}, x_{02}, x_{03}) taking the filter coefficients in pair as in the linear array; these are then collected by the second stage to give the output samples (x_{LL}, x_{HL}, x_{LH} and x_{HH}) belonging to the four sub-bands which result from the decomposition: in figure 3, outputs x_{LL} and x_{LH} are shown, while the two other samples will be computed by a different row. It is worth noting that the three sums pertaining to the high-pass filter computed in the first cell (x_{01}, x_{02}, x_{03}) feed a different input of three consecutive second stages.

4 VLSI implementation

True Single Phase Clocking logic circuits were introduced in the late 80's by Yuan and Svensson [9, 10]. This kind of circuits mixes the combinatorial logic with latch stages so that data are passed through chains of pipelined gates at a very high frequency clock. The better performance are obtained when the function to be implemented is decomposed in simple gates, with low fan-out and local interconnections. Longer interconnections, as required for example in the distribution of the control signals, must be buffered for obtaining short propagation delay.

The major drawback for this kind of logic is related to the fact that standard synthesizers cannot be employed, as they are not able to manage latched gates. Additional problems arise from the difficulty of implementing feedback loops which include pipelined stages.

The above considerations suggested to proceed with a semi-custom approach: a VHDL description of the

described architecture has been simulated and validated with standard CAD tools, then a full-custom library of the required TSPC logic cells has been developed and verified by means of electrical level simulations.

Each basic cell has been transistor size optimized to operate at the target clock frequency and then synthesized following a unique set of design rules, to help the subsequent phases of placement and routing. This approach is sub-optimal in terms of area occupation but reduces the time required for the design.

The full adder employed in the DWT processor consists of two complementary stages serially connected, as depicted in figure 4; the final layout of the full adder is drawn in figure 5, where the large metal1 clock line can be easily seen. The cell dimensions are $60 \times 150\,\mu m^2$.

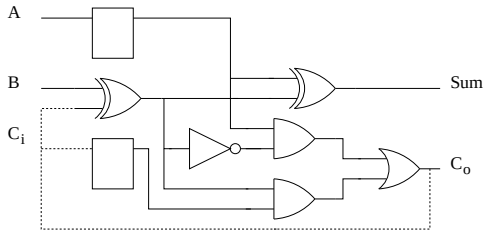

Figure 4: Gate level schematic of the full-adder

During the place and route phase some additional levels of pipelining have been added to reduce the capacitive load when local interconnections are not possible: an example is the line connecting the high-pass output of the first row filter to the corresponding input of the column filter, located at a distance of 15 rows, which means around 2.4 mm.

The developed processor performs two levels of 2d-DWT working over 32-rows image blocks. The pin-count is 160 and the silicon area is approximately $7 \times 7\,mm^2$ for a ES2 CMOS 0.7 μm technology; the post layout simulations proved the feasibility of the DWT transformation of a 512×512 grey scale picture in less than 150 μs.

5 Conclusions

The VLSI implementation of a two-dimensional Discrete Wavelet Transform processor for image coding applications has been presented.
The processor includes a linear systolic array for the row processing and a parallel array for the column filtering. The chosen I/O data parallelism allows a good tuning of the core size with the pin count of the chip. The synthesis has been performed using a True Single Phase Clocking logic library, with manual place and route: the optimization of the library cells and the fine grain pipelining of the architecture guaranty a very high working frequency (500 MHz) and performance compatible with real time applications. Future work includes the generalization of the developed structure with the aim of implementing a set of different filters.

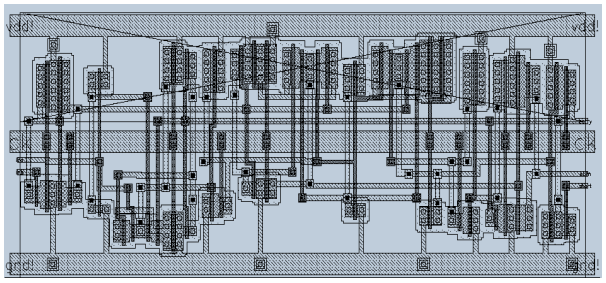

Figure 5: Layout of the full-adder TSPC

References

[1] A. S. Lewis and G. Knowles, "VLSI architecture for 2-d Daubechies wavelet transform without multipliers", *Electronic Letters*, Vol. 27, No. 2, January 1991.

[2] K.K. Parhi, T. Nishitani, "VLSI Architectures for Discrete Wavelet Transforms", *IEEE Transactions on VLSI Systems*, Vol. 1, n. 2, Jun. 1993.

[3] C. Chakrabarti and M. Vishwanath, "Efficient realizations of the discrete and continuous wavelet transform: from single chip implementations to mappings on SIMD parallel computers", *IEEE Trans. on Signal Processing*, Vol. 43, No. 3, March 1995.

[4] M. Vishwanath, R. M. Owens and M. J. Irwin, "VLSI architectures for the discrete wavelet transform", *IEEE Tans. on Circuits and Systems*, Vol. 42, No. 5, May 1995.

[5] H. Y. H. Chuang and L. Chen, "VLSI architectures for fast 2D discrete orthonormal wavelet transform", *Journal of VLSI Signal Processing*, Vol. 10, 1995.

[6] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[7] J. D. Villasenor, B. Belze and J. Liao, "Wavelet filter evaluation for image compression", *IEEE Trans. on Image Processing*, Vol. 4, No. 8, August 1995.

[8] A. Brizio, G. Masera, G. Piccinini, M. Ruo Roch, M. Zamboni, "A High Speed Bit-Serial 2d-DWT VLSI Processor", Proceedings of the *2nd IMACS International Multiconference CESA98*, Nabeul-Hammamet, Tunisia, April 1-4, 1998.

[9] J. Yuan and C. Svensson, "High speed CMOS circuit technique", *IEEE Journal of Solid-State Circuits*, Vol. 24, February 1989.

[10] J. Yuan and C. Svensson, "New single-clock CMOS latches and flip-flops with improved speed and power savings", *IEEE Journal of Solid-State Circuits*, Vol. 32, No. 1, January 1997.