REALIZATION OF MULTIWAVELET-BASED TRANSFORM KERNELS FOR IMAGE CODING

Peter Rieder, Christian V. Schimpfle and Josef A. Nossek
Institute of Network Theory & Circuit Design
Munich University of Technology
Arcisstr. 21, 80333 Munich, Germany
e-mail: chsc@nws.e-technik.tu-muenchen.de

Abstract

In this paper the efficient implementation of discrete multiwavelet transforms is examined. These transforms can be used for image coding. The presented architecture is based on lattice structures and computationally efficient CORDIC-based μ-rotations. An exemplary VLSI-implementation for a multiwavelet-based lapped orthogonal transform is presented.

1 Introduction

In recent years wavelet transforms have gained a lot of interest in many application fields, e.g. image processing. Orthogonal singlewavelets have been introduced [1], and have been generalized to orthogonal, symmetric, vector-valued multiwavelet transforms allowing more degrees of freedom [8, 5]. These multiwavelets are not only well-suited bases with respect to a multiresolution of an image, also a multiwaveletpacket transform can be constructed resulting in a lapped orthogonal transform [3] (8-channel filterbank) being a widely used transform kernel for image coding.

An efficient implementation of these transform kernels for image coding is often necessary not only because of the requirements on the processing speed, but also because of the large amount of image data. Using efficient lattice structures and coarsly quantizing their rotation angles by very few CORDIC-based μ-rotations [6, 7] leads to architectures being realizable very simple by only very few shift and add operations. Based on this strategy a very efficient VLSI-implementation for a multiwavelet-based lapped orthogonal transform is presented, whereby the area consumption can be additionally reduced by folding the architecture, by using hybrid data formats, or by time multiplexing the filter operations: Instead of one parallel 8-channel filterbank, 2 serial 4-channel filterbanks are used allowing a drastical reduction of area consumption to about $2.4mm^2$.

2 Multiwavelets

Multiwavelet systems using 2 scaling functions and 2 multiwavelets are based on 4 dilation equations

$$\Phi_v(t) = \sum_{l=1}^{2}\sum_{k=0}^{n} g_{v,l,k}\Phi_l(2t-k); \qquad v \in \{1,2\};$$

$$\Psi_v(t) = \sum_{l=1}^{2}\sum_{k=0}^{n-1} h_{v,l,k}\Phi_l(2t-k); \qquad v \in \{1,2\}.$$

and a multiwavelet basismatrix of size $4 \times 2n$

$$\mathbf{W} = \begin{pmatrix} g_{11,0} & g_{12,0} & \cdots & g_{11,n-1} & g_{12,n-1} \\ g_{21,0} & g_{22,0} & \cdots & g_{21,n-1} & g_{22,n-1} \\ h_{11,0} & h_{12,0} & \cdots & h_{11,n-1} & h_{12,n-1} \\ h_{21,0} & h_{22,0} & \cdots & h_{21,n-1} & h_{22,n-1} \end{pmatrix}$$

In order to implement the respective discrete multiwavelet transform, the filters

$$G_{v,l}(z) = \sum_{i=0}^{n-1} g_{v,l,i}z^{-i} \text{ and } H_{v,l}(z) = \sum_{i=0}^{n-1} h_{v,l,i}z^{-i}$$

with $v, l \in \{1,2\}$ are required that appear in the filterbank of Figure 1. Note, that not only the number of inputs (I) and outputs (O) is multiplied by 2, but also a prefilter is necessary.

In [8] multiwavelets were designed that allow the properties regularity, ortogonality and symmetry, simultaneously. Thereby, Φ_1 (1st row of $\boldsymbol{W}$) and Ψ_1 (3rd row of $\boldsymbol{W}$) being symmetric, and Φ_2 (2nd row of $\boldsymbol{W}$) and Ψ_2 (4th row of $\boldsymbol{W}$) being antisymmetric, requires a specially structured wavelet basismatrix $\boldsymbol{W}$ (special multifilters).

$$\boldsymbol{W} = \begin{pmatrix} a_0 & b_0 & a_1 & b_1 & a_1 & -b_1 & a_0 & -b_0 \\ -a_0 & b_0 & a_1 & -b_1 & -a_1 & -b_1 & a_0 & b_0 \\ b_0 & -a_0 & -b_1 & a_1 & -b_1 & -a_1 & b_0 & a_0 \\ b_0 & a_0 & b_1 & a_1 & -b_1 & a_1 & -b_0 & a_0 \end{pmatrix};$$

Setting $a_0 = 0.009977, a_1 = 0.697129, b_0 = b_1 = -0.083399$
results in the bases of $n = 4$ plotted in Figure 2.

0-7803-4455-3/98/$10.00 © 1998 IEEE

Authorized licensed use limited to: T U MUENCHEN. Downloaded on October 22, 2008 at 05:21 from IEEE Xplore. Restrictions apply.

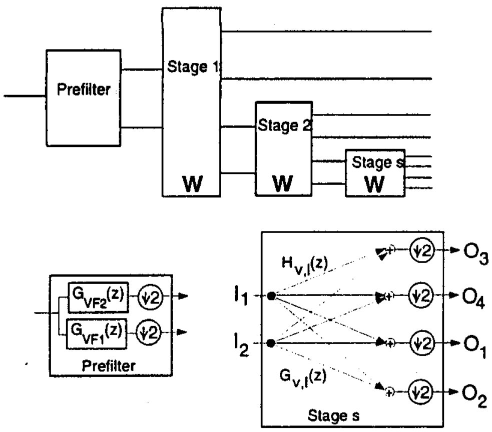

Figure 1: Filterbank for implementing vector–valued multiwavelet transforms

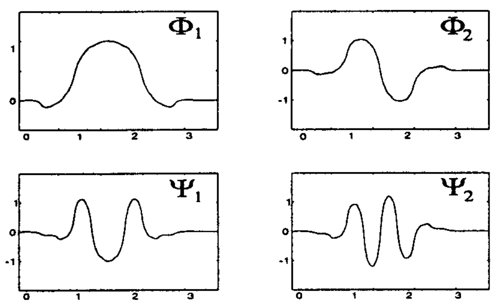

Figure 2: Multiwavelets of $n = 4, p = 2$ and the corresponding scaling functions

With respect to the implementation of the particular multifilters corresponding to these bases, a lattice structure being introduced in [5] and plotted in Figure 3 is efficient ($\alpha = 6.88^o$).

3 Efficient Implementation of Orthogonal Rotations

Since in the singlewavelet- as well as in the multiwavelet case the lattice filters are composed of orthogonal 2×2–rotations only, in order to get a simple implementation of the filters, an efficient implementation of the orthogonal rotations is sufficient.

An orthogonal 2×2–rotation $\boldsymbol{R}(\alpha)$ is defined as follows:

$$\boldsymbol{R}(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}$$

The CORDIC algorithm is a common method to execute orthogonal rotations by using a sequence of $w + 1$ μ–rotations

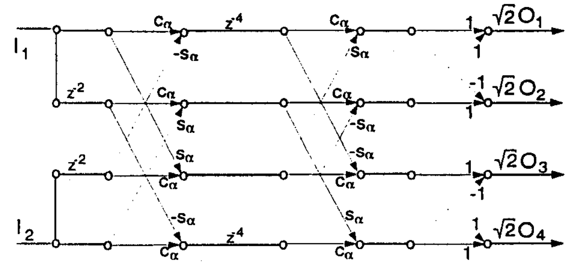

Figure 3: Lattice structure implementing multiwavelet filters of $n = 4$

(w being the wordlength):

$$\boldsymbol{R}(\alpha) = \frac{1}{K_w} \prod_{k=0}^{w} \begin{bmatrix} 1 & -\sigma_k 2^{-k} \\ \sigma_k 2^{-k} & 1 \end{bmatrix}, \quad \sigma_k \in \{+1, -1\},$$

with $\frac{1}{K_w} = \prod_{k=0}^{w} \frac{1}{\sqrt{1+2^{-2k}}}$ being the scaling factor. This corresponds to the representation of α as

$$\alpha = \sum_k \sigma_k \alpha_k = \sum_k \sigma_k \text{arctan} 2^{-k}.$$

This representation of an angle in the "$\arctan 2^{-k}$" basis is also the basic idea of CORDIC–based approximate rotations [2], but there we have $\sigma_k \in \{-1, 0, +1\}$.

In [2] double rotations consisting of 2 equal CORDIC elementary rotations were used, which rotate by the angle $2\alpha_k$, i.e. $\boldsymbol{R}(2\alpha_k) = \boldsymbol{R}(\alpha_k)\boldsymbol{R}(\alpha_k)$ such that

$$\boldsymbol{R}(2\alpha_k) = \frac{1}{K_k^2} \begin{bmatrix} 1 & -\sigma 2^{-k} \\ \sigma 2^{-k} & 1 \end{bmatrix} \begin{bmatrix} 1 & -\sigma 2^{-k} \\ \sigma 2^{-k} & 1 \end{bmatrix}$$

Now, the scaling factor $\frac{1}{K_k^2}$ can be factorized into a sequence of shift & add operations:

$$\frac{1}{K_k^2} = \frac{1}{1+2^{-2k}} = (1-2^{-2k})(1+2^{-4k})(1+2^{-8k})\dots \quad (1)$$

With one (or a few) of these double rotations an approximate rotation can be composed, that is simple to implement, approximates any rotation angle to a certain accuracy (increasing the number of double rotations increases the accuracy), and is always exactly orthogonal independent of the accuracy. For our example of Section 2, an approximation of the exact rotation angle $\alpha = 6.88$ by $\alpha \approx \arctan 2^{-3} = 7.12^o$ does not affect the quality of the transform, the corresponding bases cannot be distinguished from those of Figure 2. Thereby again double rotations are used, since in each path of the lattice filter each rotation appears twice.

Authorized licensed use limited to: T U MUENCHEN. Downloaded on October 22, 2008 at 05:21 from IEEE Xplore. Restrictions apply.

4 VLSI-Implementation of an 8-channel Multiwaveletpacket Transform

In order to get an 8-channel filterbank, the multiwavelet-packet structure of Figure 4 is used, requiring the prefilters $G_{VF1} = (1 + z^{-1})/\sqrt{2}$ and $G_{VF2} = (1 - z^{-1})/\sqrt{2}$ being equivalent to a simple 45^o–rotation. Approximating the exact

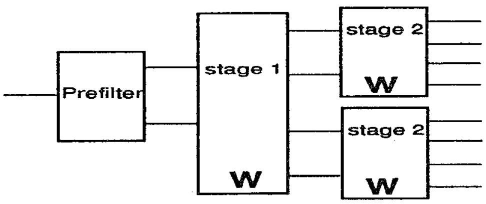

Figure 4: 8–channel multiwaveletpacket structure

rotation angles of the multifilter structure results in the architecture of Figure 5 that requires only very few shift and add operations. Note, that the scaling being necessary to normalize the analysis and synthesis part is not plotted in Figure 5, it can be put at the beginning of the analysis or at the end of the synthesis part. With this method the multiwavelet-based lap-

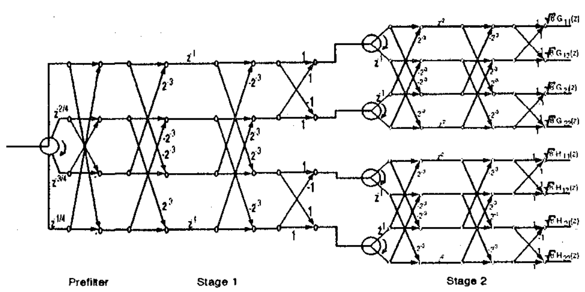

Figure 5: Architecture for efficiently implementing the multiwaveletpacket-based lapped orthogonal transform

ped orthogonal transform can be computed in a very efficient way.

The clock rate in the first stage of the filterbank is twice the clock rate in the second stage resulting in a hardware utilization of only 50% in the second stage. Now the idea is to use only one filter for the second stage and compute both, the lowpass and the highpass part within this single filter. One filter is used for the first stage producing the 4 different frequency channels, which are further processed in the second stage. The second filter stage then has to operate at the same clock rate as the first stage, where the 4 lowpass channels (of the whole 8) are computed at the odd time steps and the remaining 4 highpass channels are computed at the even time steps. For the whole filterbank working correctly, a permutation module according to Figure 6 is necessary between the two stages which works basically as a time multiplexer. With this method, full hardware utilization and a reduction of the total area of the complete filterbank by about 25% is achieved.

Figure 7 shows the complete mask layout for this architecture on a chip area of $2.4mm^2$ and a transistor density of 6307 transistors/mm^2. The pixels of the considered images have a resolution of 8 bit. To reduce quantization effects and to avoid overflow, the data word is extended to 16 bit. Due to its simple implementation, a two's complement number representation is chosen. To achieve higher throughput, the whole filterbank is pipelined at wordlevel.

Another possibility of implementation with about the same area consumption would be the use of hybrid data formats. The data word is thereby split up in a LSB and a MSB part and then represented in a block of $\frac{w}{2}$x2 bit, where w is the wordlength. Both filters in the second stage can then be implemented with only half of the hardware, as they have to compute data with a length of only $\frac{w}{2}$. Again, the second stage works at the same clock rate as the first stage. Such a bit serial approach was already presented in [9] for the implementation of a singlewavelet transform.

A third implementation method would be a wordlevel folded architecture [8], where the whole transform is executed in only one filter stage with a high throughput rate. Unfortunately, this requires an additional control circuitry and more complex interconnections than the time multiplexed version used for the layout implementation in Figure 7.

5 Conclusion

In this paper a VLSI-suited architecture for a multiwavelet-based lapped orthogonal transform is presented. It is shown that lattice structures implementing the multifilters, and CORDIC-based μ–rotations for the particular rotations allow to reduce the computational expense drastically. In order to reduce the area consumption and to achieve full hardware utilization, the principle of multiplexing is applied, which is only possible, since exactly the same multifilters appear several times in the architecture of the complete transform. With the resulting area consumption being only 2.4 mm^2, the problem of realizing transform kernels for image coding application is solved very efficiently.

References

[1] I. Daubechies. *Ten Lectures on Wavelets.* Notes from the 1990 CBMS-NSF Conference on Wavelets and Applications at Lowell, MA. SIAM, Philadelphia, PA, 1992.

[2] J. Götze and G. J. Hekstra. Adaptive Approximate Rotations for Computing the EVD. In M. Moonen and F. Cathoor, editors, *Algorithms and Parallel VLSI Architectures.* Elsevier Science Publishers, 1994.

Authorized licensed use limited to: T U MUENCHEN. Downloaded on October 22, 2008 at 05:21 from IEEE Xplore. Restrictions apply.

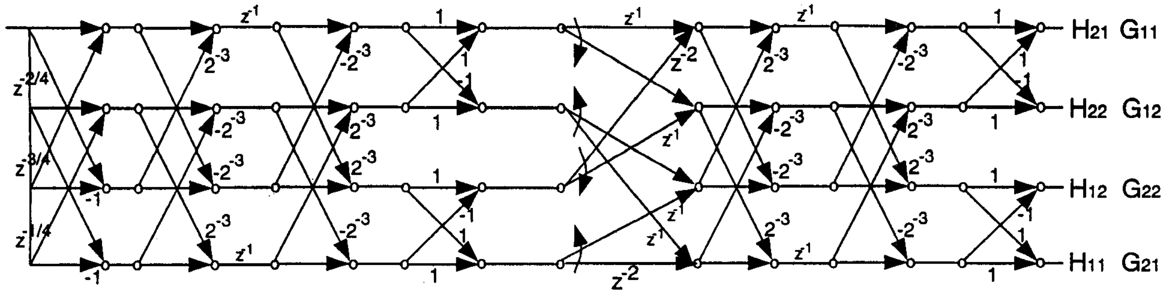

Figure 6: Multiplexed version of the 8–channel multiwavelet packet transform. The second filter stage computes the 4 lowpass channels at the odd, and the 4 highpass channels at the even time steps.

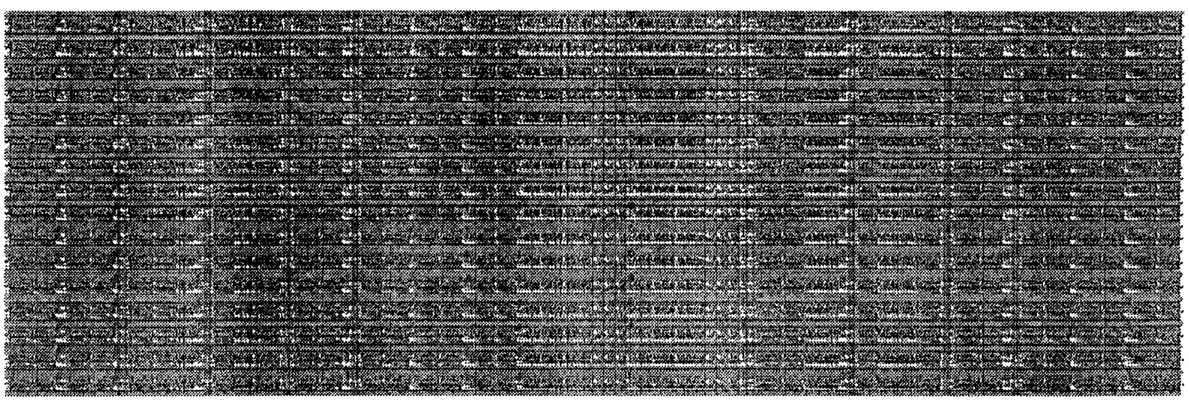

Figure 7: VLSI–implementation of the multiplexed version of the 8–channel multiwavelet packet transform

[3] H.S. Malvar and D.H. Staelin. The LOT: Transform Coding Without Blocking Effects. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 37(4):553–559, April 1989.

[4] K. K. Parhi and T. Nishitani. Folded VLSI Architectures for Discete Wavelet Transform. *Proc. ISCAS*, pages 1734–1737, 1993.

[5] P.Rieder. Parameterization of Symmetric Multiwavelets. *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing, ICASSP 97, Munich*, April 1997.

[6] P. Rieder, K. Gerganoff, J. Götze, and J.A. Nossek. Parameterization and Implementation of Orthogonal Wavelet transforms. *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing, ICASSP 96, Atlanta*, III:1515–1518, Mai 1996.

[7] P. Rieder, J. Götze, J.A. Nossek, and C.S. Burrus. Parameterization of Orthogonal Wavelet Transforms and their Simple Implementation. *IEEE Trans. on Circuits and Systems II*, 1997.

[8] P. Rieder and J.A. Nossek. Smooth Multiwavelets based on 2 Scaling Functions. *Proc. IEEE Int. Symp. on Time–Frequency and Time–Scale Analysis*, pages 309–312, June 1996. Paris.

[9] S. Simon, P. Rieder, C. Schimpfle, and J. A. Nossek. CORDIC-Based Architectures for the Efficient Implementation of Discete Wavelet Transforms. *Proc. ISCAS*, pages IV:77–80, 1996.

Authorized licensed use limited to: T U MUENCHEN. Downloaded on October 22, 2008 at 05:21 from IEEE Xplore. Restrictions apply.