

VLSI

VLSI

Edited by
ZHONGFENG WANG

In-Tech
intechweb.org

Published by In-Teh

In-Teh

Olajnica 19/2, 32000 Vukovar, Croatia

Abstracting and non-profit use of the material is permitted with credit to the source. Statements and opinions expressed in the chapters are those of the individual contributors and not necessarily those of the editors or publisher. No responsibility is accepted for the accuracy of information contained in the published articles. Publisher assumes no responsibility liability for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained inside. After this work has been published by the In-Teh, authors have the right to republish it, in whole or part, in any publication of which they are an author or editor, and the make other personal use of the work.

© 2010 In-teh

www.intechweb.org

Additional copies can be obtained from:

publication@intechweb.org

First published February 2010

Printed in India

Technical Editor: Melita Horvat

Cover designed by Dino Smrekar

VLSI,

Edited by Zhongfeng Wang

p. cm.
ISBN 978-953-307-049-0

Preface

The process of integrated circuits (IC) started its era of very-large-scale integration (VLSI) in 1970's when thousands of transistors were integrated into a single chip. Since then, the transistors counts and clock frequencies of state-of-art chips have grown by orders of magnitude. Nowadays we are able to integrate more than a billion transistors into a single device. However, the term "VLSI" remains being commonly used, despite of some effort to coin a new term ultralarge- scale integration (ULSI) for finer distinctions many years ago. In the past two decades, advances of VLSI technology have led to the explosion of computer and electronics world. VLSI integrated circuits are used everywhere in our everyday life, including microprocessors in personal computers, image sensors in digital cameras, network processors in the Internet switches, communication devices in smartphones, embedded controllers in automobiles, et al.

VLSI covers many phases of design and fabrication of integrated circuits. In a complete VLSI design process, it often involves system definition, architecture design, register transfer language (RTL) coding, pre- and post-synthesis design verification, timing analysis, and chip layout for fabrication. As the process technology scales down, it becomes a trend to integrate many complicated systems into a single chip, which is called system-on-chip (SoC) design. In addition, advanced VLSI systems often require high-speed circuits for the ever increasing demand of data processing. For instance, Ethernet standard has evolved from 10 Mbps to 10 Gbps, and the specification for 100 Gbps Ethernet is underway. On the other hand, with the growing popularity of smartphones and mobile computing devices, low-power VLSI systems have become critically important. Therefore, engineers are facing new challenges to design highly integrated VLSI systems that can meet both high performance requirement and stringent low power consumption.

The goal of this book is to elaborate the state-of-art VLSI design techniques at multiple levels. At device level, researchers have studied the properties of nano-scale devices and explored possible new material for future very high speed, low-power chips. At circuit level, interconnect has become a contemporary design issue for nano-scale integrated circuits. At system level, hardware-software co-design methodologies have been investigated to coherently improve the overall system performance. At architectural level, researchers have proposed novel architectures that have been optimized for specific applications as well as efficient reconfigurable architectures that can be adapted for a class of applications.

As VLSI systems become more and more complex, it is a great challenge but a significant task for all experts to keep up with latest signal processing algorithms and associated architecture designs. This book is to meet this challenge by providing a collection of advanced algorithms

in conjunction with their optimized VLSI architectures, such as Turbo codes, Low Density Parity Check (LDPC) codes, and advanced video coding standards MPEG4/H.264, et al. Each of the selected algorithms is presented with a thorough description together with research studies towards efficient VLSI implementations. No book is expected to cover every possible aspect of VLSI exhaustively. Our goal is to provide the design concepts through those selected studies, and the techniques that can be adopted into many other current and future applications.

This book is intended to cover a wide range of VLSI design topics – both general design techniques and state-of-art applications. It is organized into four major parts:

- Part I focuses on VLSI design for image and video signal processing systems, at both algorithmic and architectural levels.
- Part II addresses VLSI architectures and designs for cryptography and error correction coding.
- Part III discusses general SoC design techniques as well as system-level design optimization for application-specific algorithms.
- Part IV is devoted to circuit-level design techniques for nano-scale devices.

It should be noted that the book is not a tutorial for beginners to learn general VLSI design methodology. Instead, it should serve as a reference book for engineers to gain the knowledge of advanced VLSI architecture and system design techniques. Moreover, this book also includes many in-depth and optimized designs for advanced applications in signal processing and communications. Therefore, it is also intended to be a reference text for graduate students or researchers for pursuing in-depth study on specific topics.

The editors are most grateful to all coauthors for contributions of each chapter in their respective area of expertise. We would also like to acknowledge all the technical editors for their support and great help.

Zhongfeng Wang, Ph.D.
Broadcom Corp., CA, USA

Xinming Huang, Ph.D.
Worcester Polytechnic Institute, MA, USA

Contents

Preface	V
1. Discrete Wavelet Transform Structures for VLSI Architecture Design Hannu Olkkonen and Juuso T. Olkkonen	001
2. High Performance Parallel Pipelined Lifting-based VLSI Architectures for Two-Dimensional Inverse Discrete Wavelet Transform Ibrahim Saeed Koko and Herman Agustiawan	011
3. Contour-Based Binary Motion Estimation Algorithm and VLSI Design for MPEG-4 Shape Coding Tsung-Han Tsai, Chia-Pin Chen, and Yu-Nan Pan	043
4. Memory-Efficient Hardware Architecture of 2-D Dual-Mode Lifting-Based Discrete Wavelet Transform for JPEG2000 Chih-Hsien Hsia and Jen-Shiun Chiang	069
5. Full HD JPEG XR Encoder Design for Digital Photography Applications Ching-Yen Chien, Sheng-Chieh Huang, Chia-Ho Pan and Liang-Gee Chen	099
6. The Design of IP Cores in Finite Field for Error Correction Ming-Haw Jing, Jian-Hong Chen, Yan-Haw Chen, Zih-Heng Chen and Yaotsu Chang	115
7. Scalable and Systolic Gaussian Normal Basis Multipliers over GF(2 ^m) Using Hankel Matrix-Vector Representation Chiou-Yng Lee	131
8. High-Speed VLSI Architectures for Turbo Decoders Zhongfeng Wang and Xinming Huang	151
9. Ultra-High Speed LDPC Code Design and Implementation Jin Sha, Zhongfeng Wang and Minglun Gao	175
10. A Methodology for Parabolic Synthesis Erik Hertz and Peter Nilsson	199
11. Fully Systolic FFT Architectures for Giga-sample Applications D. Reisis	221

12. Radio-Frequency (RF) Beamforming Using Systolic FPGA-based Two Dimensional (2D) IIR Space-time Filters Arjuna Madanayake and Leonard T. Bruton	247
13. A VLSI Architecture for Output Probability Computations of HMM-based Recognition Systems Kazuhiro Nakamura, Masatoshi Yamamoto, Kazuyoshi Takagi and Naofumi Takagi	273
14. Efficient Built-in Self-Test for Video Coding Cores: A Case Study on Motion Estimation Computing Array Chun-Lung Hsu, Yu-Sheng Huang and Chen-Kai Chen	285
15. SOC Design for Speech-to-Speech Translation Shun-Chieh Lin, Jia-Ching Wang, Jhing-Fa Wang, Fan-Min Li and Jer-Hao Hsu	297
16. A Novel De Bruijn Based MeshTopology for Networks-on-Chip Reza Sabbaghi-Nadooshan, Mehdi Modarressi and Hamid Sarbazi-Azad	317
17. On the Efficient Design & Synthesis of Differential Clock Distribution Networks Houman Zarrabi, Zeljko Zilic, Yvon Savaria and A. J. Al-Khalili	331
18. Robust Design and Test of Analog/Mixed-Signal Circuits in Deeply Scaled CMOS Technologies Guo Yu and Peng Li	353
19. Nanoelectronic Design Based on a CNT Nano-Architecture Bao Liu	375
20. A New Technique of Interconnect Effects Equalization by using Negative Group Delay Active Circuits Blaise Ravelo, André Pérennec and Marc Le Roy	409
21. Book Embeddings Saïd Bettayeb	435
22. VLSI Thermal Analysis and Monitoring Ahmed Lakhssassi and Mohammed Bougataya	441

Discrete Wavelet Transform Structures for VLSI Architecture Design

Hannu Olkkonen and Juuso T. Olkkonen

Department of Physics, University of Kuopio, 70211 Kuopio, Finland

VTT Technical Research Centre of Finland, 02044 VTT, Finland

1. Introduction

Wireless data transmission and high-speed image processing devices have generated a need for efficient transform methods, which can be implemented in VLSI environment. After the discovery of the compactly supported discrete wavelet transform (DWT) (Daubechies, 1988; Smith & Barnwell, 1986) many DWT-based data and image processing tools have outperformed the conventional discrete cosine transform (DCT) -based approaches. For example, in JPEG2000 Standard (ITU-T, 2000), the DCT has been replaced by the biorthogonal discrete wavelet transform. In this book chapter we review the DWT structures intended for VLSI architecture design. Especially we describe methods for constructing shift invariant analytic DWTs.

2. Biorthogonal discrete wavelet transform

The first DWT structures were based on the compactly supported conjugate quadrature filters (CQFs) (Smith & Barnwell, 1986), which had nonlinear phase effects such as image blurring and spatial dislocations in multi-resolution analyses. On the contrary, in biorthogonal discrete wavelet transform (BDWT) the scaling and wavelet filters are symmetric and linear phase. The two-channel analysis filters $H_0(z)$ and $H_1(z)$ (Fig. 1) are of the general form

$$\begin{aligned} H_0(z) &= (1 + z^{-1})^K P(z) \\ H_1(z) &= (1 - z^{-1})^K Q(z) \end{aligned} \quad (1)$$

where the scaling filter $H_0(z)$ has the Kth order zero at $\omega = \pi$. The wavelet filter $H_1(z)$ has the Kth order zero at $\omega = 0$, correspondingly. $P(z)$ and $Q(z)$ are polynomials in z^{-1} . The reconstruction filters $G_0(z)$ and $G_1(z)$ (Fig. 1) obey the well-known perfect reconstruction condition

$$\begin{aligned} H_0(z)G_0(z) + H_1(z)G_1(z) &= 2z^{-k} \\ H_0(-z)G_0(z) + H_1(-z)G_1(z) &= 0 \end{aligned} \quad (2)$$

The last condition in (2) is satisfied if we select the reconstruction filters as $G_0(z) = H_1(-z)$ and $G_1(z) = -H_0(-z)$.

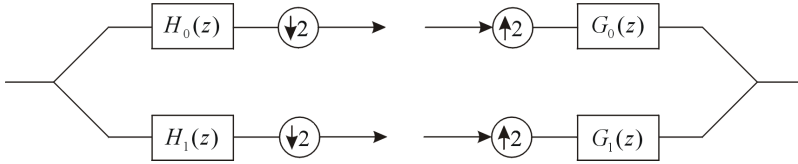


Fig. 1. Analysis and synthesis BDWT filters.

3. Lifting BDWT

The BDWT is most commonly realized by the ladder-type network called lifting scheme (Sweldens, 1988). The procedure consists of sequential down and uplifting steps and the reconstruction of the signal is made by running the lifting network in reverse order (Fig. 2). Efficient lifting BDWT structures have been developed for VLSI design (Olkkonen et al. 2005). The analysis and synthesis filters can be implemented by integer arithmetics using only register shifts and summations. However, the lifting DWT runs sequentially and this may be a speed-limiting factor in some applications (Huang et al., 2005). Another drawback considering the VLSI architecture is related to the reconstruction filters, which run in reverse order and two different VLSI realizations are required. In the following we show that the lifting structure can be replaced by more effective VLSI architectures. We describe two different approaches: the discrete lattice wavelet transform and the sign modulated BDWT.

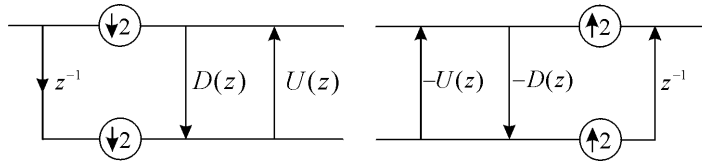


Fig. 2. The lifting BDWT structure.

4. Discrete lattice wavelet transform

In the analysis part the discrete lattice wavelet transform (DLWT) consists of the scaling $H_0(z)$ and wavelet $H_1(z)$ filters and the lattice network (Fig. 3). The lattice structure contains two parallel transmission filters $T_0(z)$ and $T_1(z)$, which exchange information via two crossed lattice filters $L_0(z)$ and $L_1(z)$. In the synthesis part the lattice structure consists of the transmission filters $R_0(z)$ and $R_1(z)$ and crossed filters $W_0(z)$ and $W_1(z)$, and finally the reconstruction filters $G_0(z)$ and $G_1(z)$. Supposing that the scaling and wavelet filters obey (1), for perfect reconstruction the lattice structure should follow the condition

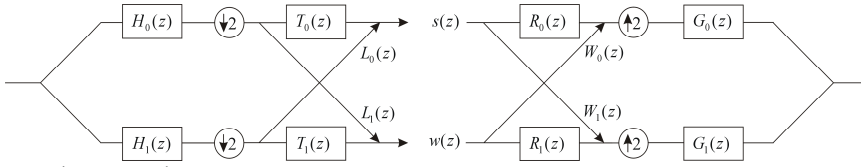


Fig. 3. The general DLWT structure.

$$\begin{bmatrix} T_0R_0 + L_1W_0 & L_0R_0 + T_1W_0 \\ T_0W_1 + L_1R_1 & T_1R_1 + L_0W_1 \end{bmatrix} = \begin{bmatrix} z^{-k} & 0 \\ 0 & z^{-k} \end{bmatrix} \quad (3)$$

This is satisfied if we state $W_0 = -L_0$, $W_1 = -L_1$, $R_0 = T_1$ and $R_1 = T_0$. The perfect reconstruction condition follows then from the diagonal elements (3) as

$$T_0(z)T_1(z) - L_0(z)L_1(z) = z^{-k} \quad (4)$$

There exists many approaches in the design of the DLWT structures obeying (4), for example via the Parks-McChellan-type algorithm. Especially the DLWT network is efficient in designing half-band transmission and lattice filters (see details in Olkkonen & Olkkonen, 2007a). For VLSI design it is essential to note that in the lattice structure all computations are carried out parallel. Also all the BDWT structures designed via the lifting scheme can be transferred to the lattice network (Fig. 3). For example, Fig. 4 shows the DLWT equivalent of the lifting DBWT structure consisting of down and uplifting steps (Fig. 2). The VLSI implementation is flexible due to parallel filter blocks in analysis and synthesis parts.

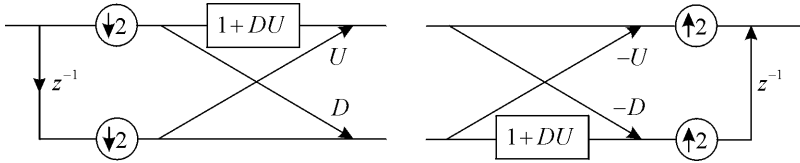


Fig. 4. The DLWT equivalence of the lifting BDWT structure described in Fig. 2.

5. Sign modulated BDWT

In VLSI architectures, where the analysis and synthesis filters are directly implemented (Fig. 1), the VLSI design simplifies considerably using a specific sign modulator defined as (Olkkonen & Olkkonen 2008)

$$S_n = (-1)^n = \begin{cases} 1 & \text{for } n \text{ even} \\ -1 & \text{for } n \text{ odd} \end{cases} \quad (5)$$

A key idea is to replace the reconstruction filters by scaling and wavelet filters using the sign modulator (5). Fig. 5 describes the rules how $H(-z)$ can be replaced by $H(z)$ and the sign modulator in connection with the decimation and interpolation operators. Fig. 6

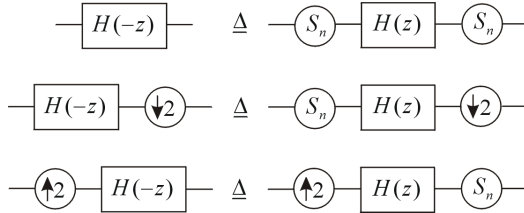


Fig. 5. The equivalence rules applying the sign modulator.

describes the general BDWT structure using the sign modulator. The VLSI design simplifies to the construction of two parallel biorthogonal filters and the sign modulator. It should be pointed out that the scaling and wavelet filters can be still efficiently implemented using the lifting scheme or the lattice structure. The same biorthogonal DWT/IDWT filter module can be used in decomposition and reconstruction of the signal e.g. in video compression unit. Especially in bidirectional data transmission the DWT/IDWT transceiver has many advantages compared with two separate transmitter and receiver units. The same VLSI module can also be used to construct multiplexer-demultiplexer units. Due to symmetry of the scaling and wavelet filter coefficients a fast convolution algorithm can be used for implementation of the filter modules (see details Olkkonen & Olkkonen, 2008).

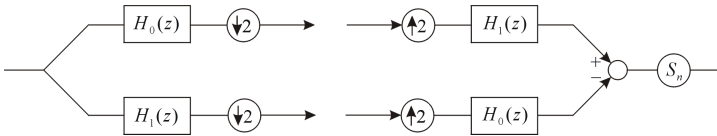


Fig. 6. The BDWT structure using the scaling and wavelet filters and the sign modulator.

6. Design example: Symmetric half-band wavelet filter for compression coder

The general structure for the symmetric half-band filter (HBF) is, for k odd

$$H(z) = z^{-k} + B(z^2) \quad (6)$$

where $B(z^2)$ is a symmetric polynomial in z^{-2} . The impulse response of the HBF contains only one odd point. For example, we may parameterize the eleven point HBF impulse response as $h[n] = [c \ 0 \ b \ 0 \ a \ 1 \ a \ 0 \ b \ 0 \ c]$, which has three adjustable parameters. The compression efficiency improves when the high-pass wavelet filter approaches the frequency response of the sinc-function, which has the HBF structure. However, the impulse response of the sinc-function is infinite, which prolongs the computation time. In this work we select the seven point compactly supported HBF prototype as a wavelet filter, which has the impulse response

$$h_i[n] = [b \ 0 \ a \ 1 \ a \ 0 \ b] \quad (7)$$

containing two adjustable parameters a and b . In our previous work we have introduced a modified regulatory condition for computation of the parameters of the wavelet filter (Olkkonen et al. 2005)

$$\sum_{n=0}^N n^m h_i[n] = 0; m = 0, 1, \dots, M-1 \quad (8)$$

This relation implies that $H_1(z)$ contains Mth-order zero at $z=1$ ($\omega=0$), where M is the number of vanishing moments. Writing (8) for the prototype filter (7) we obtain two equations $2a+2b+1=0$ and $20a+36b+9=0$, which give the solution $a=-9/16$ and $b=1/16$. The wavelet filter has the z-transform

$$H_1(z) = (1-z^{-1})^4(1+4z^{-1}+z^{-2})/16 \tag{9}$$

having fourth order root at $z=1$. The wavelet filter can be realized in the HBF form

$$H_1(z) = z^{-3} - A(z^2); A(z^2) = (-1+9z^{-2}+9z^{-4}-z^{-6})/16 \tag{10}$$

Using the equivalence

$$[H(z^2)]_{\downarrow 2} \equiv (\downarrow 2)H(z) \tag{11}$$

the HBF structure can be implemented using the lifting scheme (Fig. 7). The functioning of the compression coder can be explained by writing the input signal via the polyphase components

$$X(z) = X_e(z^2) + z^{-1}X_o(z^2) \tag{12}$$

where $X_e(z)$ and $X_o(z)$ denote the even and odd sequences. We may present the wavelet coefficients as

$$W(z) = [X(z)H_1(z)]_{\downarrow 2} = z^{-2}X_o(z) - A(z)X_e(z) \tag{13}$$

$A(z)$ works as an approximating filter yielding an estimate of the odd data points based on the even sequence. The wavelet sequence $W(z)$ can be interpreted as the difference between the odd points and their estimate. In tree structured compression coder the scaling sequence $S(z)$ is fed to the next stage. In many VLSI applications, for example image compression, the input signal consists of an integer-valued sequences. By rounding or truncating the output of the $A(z)$ filter to integers, the compressed wavelet sequence $W(z)$ is integer-valued and can be efficiently coded e.g. using Huffman algorithm. It is essential to note that this integer-to-integer transform has still the perfect reconstruction property (2).

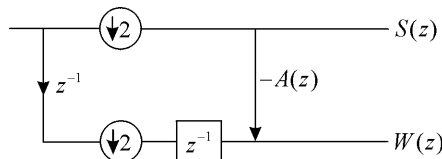


Fig. 7. The lifting structure for the HBF wavelet filter designed for the VLSI compression coder.

7. Shift invariant BDWT

The drawback in multi-scale BWDT analysis of signals and images is the dependence of the total energy of the wavelet coefficients on the fractional shifts of the analysed signal. If we have a discrete signal $x[n]$ and the corresponding time shifted signal $x[n-\tau]$, where $\tau \in [0,1]$, there may exist a significant difference in the energy of the wavelet coefficients as a function of the time shift. Kingsbury (2001) proposed a nearly shift invariant complex wavelet transform, where the real and imaginary wavelet coefficients are approximately Hilbert transform pairs. The energy (absolute value) of the wavelet coefficients equals the envelope,

which warrants smoothness and shift invariance. Selesnick (2002) observed that using two parallel CQF banks, which are constructed so that the impulse responses of the scaling filters are half-sample delayed versions of each other: $h_0[n]$ and $h_0[n-0.5]$, the corresponding wavelets are Hilbert transform pairs. In z-transform domain we should be able to construct the scaling filters $H_0(z)$ and $z^{-0.5}H_0(z)$. However, the constructed scaling filters do not possess coefficient symmetry and in multi-scale analysis the nonlinearity disturbs spatial timing and prevents accurate statistical correlations between different scales. In the following we describe the shift invariant BDWT structures especially designed for VLSI applications.

7.1 Half-delay filters for shift invariant BDWT

The classical approach for design of the half-sample delay filter $D(z)$ is based on the Thiran all-pass interpolator

$$D(z) = z^{-0.5} = \prod_{k=1}^p \frac{c_k + z^{-1}}{1 + c_k z^{-1}} \quad (14)$$

where the c_k coefficients are designed so that the frequency response follows approximately

$$D(\omega) = e^{-j\omega/2} \quad (15)$$

Recently, half-delay B-spline filters have been introduced, which have an ideal phase response. The method yields linear phase and shift invariant transform coefficients and can be adapted to any of the existing BDWT (Olkkonen & Olkkonen, 2007b). The half-sample delayed scaling and wavelet filters and the corresponding reconstruction filters are

$$\begin{aligned} \bar{H}_0(z) &= D(z)H_0(z) \\ \bar{H}_1(z) &= D^{-1}(-z)H_1(z) \\ \bar{G}_0(z) &= D^{-1}(z)G_0(z) \\ \bar{G}_1(z) &= D(-z)G_1(z) \end{aligned} \quad (16)$$

The half-delayed BDWT filter bank obeys the perfect reconstruction condition (2). The B-spline half-delay filters have the IIR structure

$$D(z) = \frac{A(z)}{B(z)} \quad (17)$$

which can be implemented by the inverse filtering procedure (see details Olkkonen & Olkkonen 2007b).

7.2 Hilbert transform-based shift invariant DWT

The tree-structured complex DWT is based on the FFT-based computation of the Hilbert transform (H_s operator in Fig. 8). The scaling and wavelet filters both obey the HBF structure (Olkkonen et al. 2007c)

$$\begin{aligned} H_0(z) &= \frac{1}{2} + z^{-1}B(z^2) \\ H_1(z) &= \frac{1}{2} - z^{-1}B(z^2) \end{aligned} \quad (18)$$

For example, the impulse response $h_0[n] = [-1 \ 0 \ 9 \ 16 \ 9 \ 0 \ -1]/32$ has the fourth order zero at $\omega = \pi$ and $h_1[n] = [1 \ 0 \ -9 \ 16 \ -9 \ 0 \ 1]/32$ has the fourth order zero at $\omega = 0$. In the tree structured HBF DWT the wavelet sequences $w_a[n]$. A key feature is that the odd coefficients of the analytic signal $w_a[2n+1]$ can be reconstructed from the even coefficient values $w_a[2n]$. This avoids the need to use any reconstruction filters. The HBFs (18) are symmetric with respect to $\omega = \pi/2$. Hence, the energy in the frequency range $0 \rightarrow \pi$ is equally divided by the scaling and wavelet filters and the energy (absolute value) of the scaling and wavelet

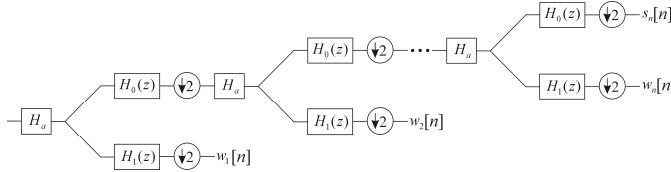


Fig. 8. Hilbert transform-based shift invariant DWT.

coefficients are statistically comparable. The computation of the analytic signal via the Hilbert transform requires the FFT-based signal processing. However, efficient FFT chips are available for VLSI implementation. In many respects the advanced method outperforms the previous nearly shift invariant DWT structures.

7.3 Hilbert transform filter for construction of shift invariant BDWT

The FFT-based implementation of the shift invariant DWT can be avoided if we define the Hilbert transform filter $\mathcal{H}(z)$, which has the frequency response

$$\mathcal{H}(\omega) = e^{-j\pi/2} \text{sgn}(\omega) \quad (19)$$

where $\text{sgn}(\omega) = 1$ for $\omega \geq 0$ and $\text{sgn}(\omega) = -1$ for $\omega < 0$. In the following we describe a novel method for constructing the Hilbert transform filter based on the half-sample delay filter $D(z)$ (17), whose frequency response follows (15). The quadrature mirror filter $D(-z)$ has the frequency response

$$D(\omega - \pi) = e^{-j(\omega - \pi)/2} \quad (20)$$

The frequency response of the filter $D(z)D^{-1}(-z)$ is, correspondingly

$$\frac{D(\omega)}{D(\omega - \pi)} = e^{-j\omega/2} e^{j(\omega - \pi)/2} = e^{-j\pi/2} \quad (21)$$

Comparing (19) and using notation (17) we obtain the Hilbert transform filter as

$$\mathcal{H}(z) = \frac{A(z)B(-z)}{A(-z)B(z)} \quad (22)$$

The corresponding parallel BDWT filter bank is

$$\begin{aligned} \bar{H}_0(z) &= \mathcal{H}(z)H_0(z) \\ \bar{H}_1(z) &= \mathcal{H}^{-1}(-z)H_1(z) \\ \bar{G}_0(z) &= \mathcal{H}^{-1}(z)G_0(z) \\ \bar{G}_1(z) &= \mathcal{H}(-z)G_1(z) \end{aligned} \quad (23)$$

By filtering the real-valued signal $X(z)$ by the Hilbert transform filter results in an analytic signal $[1 + j\mathcal{H}(z)]X(z)$, whose magnitude response is zero at negative side of the frequency spectrum. For example, an integer-valued half-delay filter $D(z)$ for this purpose is obtained by the B-spline transform (Olkkonen & Olkkonen, 2007b). The frequency response of the Hilbert transform filter designed by the fourth order B-spline (Fig. 9) shows a maximally flat magnitude spectrum. The phase spectrum corresponds to the ideal Hilbert transformer (19).

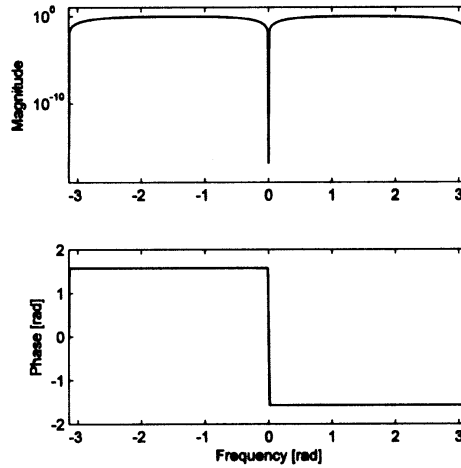


Fig. 9. Magnitude and phase spectra of the Hilbert transform filter yielded by the fourth order B-spline transform.

8. Conclusion

In this book chapter we have described the BDWT constructions especially tailored for VLSI environment. Most of the VLSI designs in the literature are focused on the biorthogonal 9/7 filters, which have decimal coefficients and usually implemented using the lifting scheme (Sweldens, 1988). However, the lifting BDWT needs two different filter banks for analysis and synthesis parts. The speed of the lifting BDWT is also limited due to the sequential lifting steps. In this work we showed that the lifting BDWT can be replaced by the lattice structure (Olkkonen & Olkkonen, 2007a). The two-channel DLWT filter bank (Fig. 3) runs parallel, which significantly increases the channel throughput. A significant advantage compared with the previous maximally decimated filter banks is that the DLWT structure allows the construction of the half-band lattice and transmission filters. In tree structured wavelet transform half-band filtered scaling coefficients introduce no aliasing when they are fed to the next scale. This is an essential feature when the frequency components in each scale are considered, for example in electroencephalography analysis.

The VLSI design of the BDWT filter bank simplifies essentially by implementing the sign modulator unit (Fig. 5), which eliminates the need for constructing separate reconstruction filters. The biorthogonal DWT/IDWT transceiver module uses only two parallel filter structures. Especially in bidirectional data transmission the DWT/IDWT module offers several advantages compared with the separate transmit and receive modules, such as the

reduced size, low power consumption, easier synchronization and timing requirements. For the VLSI designer the DWT/IDWT module appears as a "black box", which readily fits to the data under processing. This may override the relatively big barrier from the wavelet theory to the practical VLSI and microprocessor applications. As a design example we described the construction of the compression coder (Fig. 7), which can be used to compress integer-valued data sequences, e.g. produced by the analog-to-digital converters.

It is well documented that the real-valued DWTs are not shift invariant, but small fractional time-shifts may introduce significant differences in the energy of the wavelet coefficients. Kingsbury (2001) showed that the shift invariance is improved by using two parallel filter banks, which are designed so that the wavelet sequences constitute real and imaginary parts of the complex analytic wavelet transform. The dual-tree discrete wavelet transform (DT-DWT) has been shown to outperform the real-valued DWT in a variety of applications such as denoising, texture analysis, speech recognition, processing of seismic signals and neuroelectric signal analysis (Olkkonen et al. 2006). Selesnick (2002) made an observation that a half-sample time-shift between the scaling filters in parallel CQF banks is enough to produce the analytic wavelet transform, which is nearly shift invariant. In this work we described the shift invariant DT-BDWT bank (16) based on the half-sample delay filter. It should be pointed out that the half-delay filter approach yields wavelet bases which are Hilbert transform pairs, but the wavelet sequences are only approximately shift invariant. In multi-scale analysis the complex wavelet sequences should be shift invariant. This requirement is satisfied in the Hilbert transform-based approach (Fig. 8), where the signal in every scale is Hilbert transformed yielding strictly analytic and shift invariant transform coefficients. The procedure needs FFT-based computation (Olkkonen et al. 2007c), which may be an obstacle in many VLSI realizations. To avoid this we described a Hilbert transform filter for constructing the shift invariant DT-BDWT bank (23). Instead of the half-delay filter bank approach (16) the perfect reconstruction condition (2) is attained using the IIR-type Hilbert transform filters, which yield analytic wavelet sequences.

9. References

- Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Commun. Pure Appl. Math.*, Vol. 41, 909-996.
- Huang, C.T., Tseng, O.O. & Chen, L.G. (2005). Analysis and VLSI architecture for 1-D and 2-D discrete wavelet transform. *IEEE Trans. Signal Process.* Vol. 53, No. 4, 1575-1586.
- ITU-T (2000) Recommend. T.800-ISO DCD15444-1: *JPEG2000 Image Coding System*. International Organization for Standardization, ISO/IEC JTC1 SC29/WG1.
- Kingsbury, N.G. (2001). Complex wavelets for shift invariant analysis and filtering of signals. *J. Appl. Comput. Harmonic Analysis*. Vol. 10, 234-253.
- Olkkonen, H., Pesola, P. & Olkkonen, J.T. (2005). Efficient lifting wavelet transform for microprocessor and VLSI applications. *IEEE Signal Process. Lett.* Vol. 12, No. 2, 120-122.
- Olkkonen, H., Pesola, P., Olkkonen, J.T. & Zhou, H. (2006). Hilbert transform assisted complex wavelet transform for neuroelectric signal analysis. *J. Neuroscience Meth.* Vol. 151, 106-113.
- Olkkonen, J.T. & Olkkonen, H. (2007a). Discrete lattice wavelet transform. *IEEE Trans. Circuits and Systems II*. Vol. 54, No. 1, 71-75.

Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

