RESEARCH ARTICLE OPEN ACCESS

Efficient VLSI Architecture For CSD Basedsub-Band Tree Structure Using 4-Tap Filter

Radhe Kant Mishra, Dr. Subbaratnam Kumar

Abstract
A sub-band tree structure hardware design based on canonic signed digit (CSD) architecture is presented in this paper. We have proposed based on canonic signed digit (CSD) arithmetic for low complexity and efficient implementation of sub-band tree structure. The canonic signed digit (CSD) technique has been applied to reduce the number of full adders required by 2's complement based deigns. This architecture is suitable for high speed on-line applications. With this architecture the speed of the sub-band tree structure is increased with a factor two but the occupied area of the circuit is less than double. It has 100% hardware utilization efficiency.

Keywords-Sub-band tree structure (SBTS), One, Two, Three Level, Canonic signed digit (CSD) scheme, Xilinx simulation.

I. Introduction

Sub-bands, based on the time-scaling representations provide an alternative to the time-frequency representation in signal processing domain. The shifting (or translation) and the scaling (or dilation) are unique to sub-bands. The sub-band is a kind of bases which are generated by dilation and translation of a function [1], [2]. The sub-band analysis method has a good ability at localizing signal in both time and frequency plane[4].Due to the characteristic of flexible TF decomposition, sub-band tree structure has also been widely used in many applications, especially in image and video coding, speech and audio coding, speech enhancement, speech recognition, hearing aid and digital commutation [2-4].

In this paper, in the simplest form, the bit-level multiplication of two number can be performed by shift and add operation. It has been observed that the complexity of a shift-add type signed multiplier is depends on the number of ones of the 2's complement representation of the multiplicand number with the shifted partial sum whereas the zeros will only shift the partial sum. It is assumed that the shifting does not required any hardware as it can be done by hardwiring. The number of ones of the 2's complement number,therefore, will determine the numbers of full adder (FA)required implementing the multiplier. Canonic signed digit(CSD) are popular for representing a number with fewestnumber of non-zero digit.

No 2 consecutive bits in a CSD number are non-zero.

The CSD representation of a number contains the minimum possible number of nonzero bits, thus the name canonic.

The CSD representation of a number is unique.

CSD numbers cover the range (-4/3, 4/3), out of which the value in the range {-1, 1} are of greatest interest.

The CSD number has 33% fewer average non-zero bits than 2's complement number.

In this paper, we have introduced a new architecture for the sub-band tree structure usingmultiplier based architecture (MBA) and canonic signed digit based architecture (CSDBA). The algorithm for the sub-band tree structure of is analyzed in the section II. The low complexity design for sub-band tree structure in the section III. CSD based architecture for SBTS in the section IV. Proposed architecture designed in the section V. Simulation result and conclusion in the section VI and VII**.**

II. Sub-band tree structure

The model used in [5] to implement the tree structure of sub-band tree structure (SBTS) is based on the filtering process. Figure 1 depicted a complete 3-level sub-band tree structure. In this figure G and H is the high pass and low pass filter respectively.

In general, the computation period is M=8 for a j-level SBTS. The period of the 3-level computation is 8. Figure 1, The Sub band Coding Algorithm As an example, suppose that the original signal X[n] has N- sample points, spanning a frequency band of zero to π rad/s. At the first decomposition level, the signal passed through the high pass and low pass filters, followed by subsampling by 2. The output of the high pass filter has N/2- sample points (hence half the time resolution) but it only spans the frequencies □/2 to □ rad/s (hence double the frequency resolution).

The output of the low-pass filer also has N/2- sample points, but it spans the other half of the frequency band, frequencies from 0 to □/2 rad/s.

Again low and high-pass filter output passed through the same low pass and high pass filters for further decomposition. The output of the second low pass filter followed by sub sampling has N/4 samples spanning a frequency band of 0 to □ /4 rad/s, and the output of the second high pass filter followed by sub sampling has N/4 samples spanning a frequency band of □ /4 to □ /2 rad/s. The second high pass filtered signal constitutes the second level of SBTS coefficients. This signal has half the time resolution, but twice the frequency resolution of the first level signal. This process continues until two samples are left. For this specific example there would be 3 levels of decomposition, each having half the number of samples of the previous level.

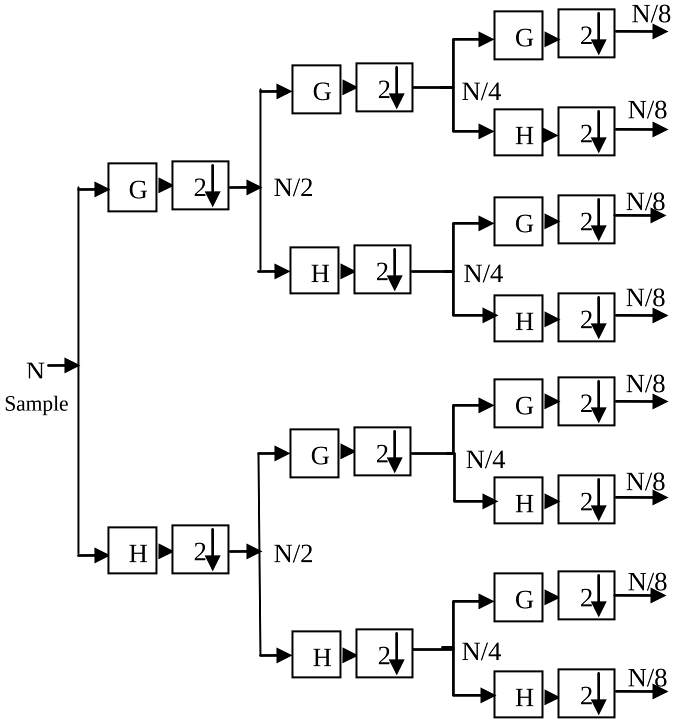

Figure 1: 3- Levels for SBTS. Where G, H are the high-pass and low-pass filter coefficient.

The SBTS of the original signal is then obtained by concatenating all coefficients starting from the last level of decomposition (remaining two samples, in this case). The SBTS will then have the same number of coefficients as the original signal.

III. LOW-COMPLEXITY DESIGNS FOR DWPT

SBTS computation is nothing but two-channel FIR filter computation. Low-pass and high-pass down sampled filter computations are performed on the input to calculate the SBTS coefficients. Low-pass down sampled filter is the average between two samples and high-pass filter is the difference b/w two samples. The SBTS algorithms for 1-level decomposition are given as;

$$Y_{high}[k] = \sum_n h[n] * x[2k-n] \quad (1)$$
$$Y_{low}[k] = \sum_n g[n] * x[2k-n] \quad (2)$$

Where x(n) is the input and $Y_{high}[k]$ & $Y_{low}[k]$ are respectively the low-pass and high-pass SBTS coefficients, h[n] and g[n] are respectively, the low-pass and high-pass filter coefficients. We have assumed the Daubechies four tap (Daub-4) filter coefficients for the low-pass filter proposed design. However, similar type of design can be derived for other type of sub-band filters as well. The Daub-4 low-pass filter coefficients are taken from [7]. The corresponding high-pass filter coefficients are calculated using the following relation:

$$g(n) = (-1)^k h(N-n)$$
(3)

Table1: Low and high-pass Daub-4 filter coefficients. The binary digit **1** shown in bold face represents -1 of the CSD digit

h(0)	0.4829629131	0.01111011	0.10000**1**0**1**
h(1)	0.8365163037	0.11010110	0.00**1**0**1**0**1**0
h(2)	0.2241438680	0.00111001	0.0100**1**001
h(3)	-0.129409522	1.11011111	1.00**1**0000**1**
g(0)	-0.129409522	1.11011111	1.00**1**0000**1**
g(1)	-0.224143868	1.11000111	1.0**1**00100**1**
g(2)	0.836516303	0.11010110	0.00**1**0**1**0**1**0
g(3)	-0.482962913	1.10000101	1.**1**0000101

Where, h(n) and g(n) are, respectively, the low and high-pass filter coefficients. N is the filter order. The 8 bit 2'complement and CSD representation of the low and high-pass filter coefficient are given in table1. Equation can be rewritten four-tap FIR filter as:

$$Y_h[k] = [h(0) + h(1)Z^{-1} + h(2)Z^{-2}h(3)Z^{-3}]\, X(n) \qquad (4)$$

$$Y_l[k] = [g(0) + g(1)Z^{-1} + g(2)Z^{-2}g(3)Z^{-3}]\, X(n) \qquad (5)$$

Where Z^{-1} operator represents one sample delay in Z-domain.

IV. CSD BASED DESIGNED FOR SBTS

Each of the multiplier unit is replaced with shifters and adders/subtraction for CSD implementation of SBTS. The constant multiplication factors of [5] are replaced with shift and adder/subtraction operation and rewritten as

Low pass filter

Y_h [k] = [x(n)>>1 – x(n)>>6 – x(n)>>8] + [-x(n-1) >> 3 –
x(n-1)>>5 – x(n-1)>>7] + [x(n-2)>>2 –x(n-2)>>5 + x(n)>>8] + [- x(n-3)>>3 – x(n-3)>>7] (6)

High pass filter

Y_g[k] = [-x(n)>>3 – x(n)>>7] + [-x(n-1)>>2 + x(n-1)>>5 – x(n-1)>>7] + [-x(n-2)>>3 – x(n-2)>>5 – x(n-
2)>>7] + [-x(n-3)>>1 + x(n-3)>>6] (7)

V. PROPOSED ARCHITECTURE

In this paper, the original signal X[n] has N-sample points, is passed through 1×2 demultipler. When select line is 0 then we get even sample and when select line is 1 then we get odd sample. After that we have passed these samples through CSD based low-pass filter, same process with high-pass filter.

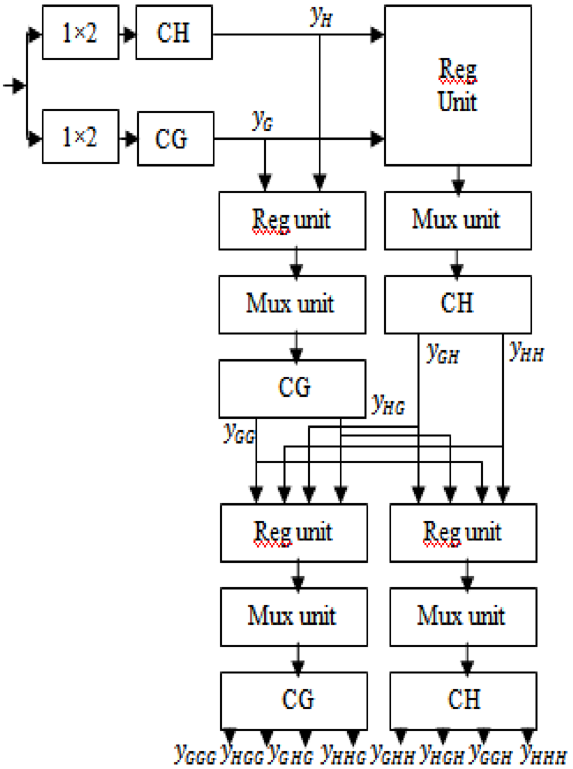

Figure 2: 3-Level CSD design based SBTS, CG and CH means the CSD design based low & high-pass filter.

Now we get N/2 sample s at the first decomposition level output of CSD based high-pass (Y_H) and low-pass filter (Y_L). At the second decomposition level, the output of CSD based low-pass and high-pass filter passed through a register unit. Now the output of register unit passed throughmux. When the select line 0, we get CSD based low-pass filter output and when the select line 1, we get CSD based high-pass filter. Now we have passed mux output through CSD based low-pass filter then we get Y_{LL}&Y_{LH} output now same process applied with the CSD based high-pass filter we get Y_{HL}&Y_{HH}. At the third decomposition level, the time period is doubled and frequency will be half, and the output of CSD based low-pass and high-pass filter is passed through a register unit. Now the output of register unit is passed through mux. When the select line is 00, we get CSD based low-pass filter output Y_{LL}, the select line is 01, we get Y_{LH} , the select line is 10 we get Y_{HL} and the select line is 11 we get Y_{HH}. Now finally we have passed mux output through CSD based low pass filter and high pass filter we get Y_{LLL}, Y_{LHL}, Y_{HLL}, Y_{HHL} and Y_{LLH}, Y_{LHH}, Y_{HLH}, Y_{HHH}.

VI. SIMULATION RESULT

We have implemented canonic signed digit based design for SBTS by different approaches. We have simulated this architecture in Xilinx 6.2i, design compiler to synthesize the design into gate level. The result is shown in Table-2 and Table-3.Table-2 shows the multiplier based SBTS and Table-2 shows the CSD based design.

Table 2: Multiplier based technique

Decomposition Level	Multiplier	Adder	No. Of slices	No. Of 4 LUTs	Required time(nsec)
First	8	6	136	247	11.129
second	16	12	355	654	18.345
Third	24	18	613	1114	25.568

Table 3: Proposed architecture (CSD based technique)

Decomposition Level	Multiplier	Adder	No. Of slices	No. Of 4 LUTs	Required time(nsec)
First	-	30	49	89	8.920
second	-	60	67	121	10.565
Third	-	90	141	252	13.018

In multiplier based technique a no. of multiplier is used while in CSD based technique the no. of multiplier is zeros similarly the number of slices is more than the CSD based technique and time is decreased. So the area and power requirement is also reduced. Here the comparison shows first level decomposition to second level decomposition and second level to third level decomposition; the adder, multiplier, no. of slices, no. of LUTs and time are reduced in CSD based technique. And if no. of slices is less then area will be reduced and if area is reduced then automatically time will be also reduced. This is the main advantage of proposed architecture.

VII. CONCLUSION

In this paper we have proposed multiplier based architecture (MBA) and canonic signed digit based architecture (CSDBA) for sub-band tree structure (SBTS). We have used CSD number system to represent the filter coefficients of thesub-band filter with minimum number of ones. Consequently, the number of FAs of the design will be reduced by nearly 50% of these of the 2’s complement design. Then we applied the CSD technique to further reduce the power and area. In this architecture the speed of the input sampling increased and used of the low and high pass filter. Low pass filter is the average between two sample and high pass filter is the difference between two samples. There is no on-chip memory and memory access during the computation, so that can achieve significant reduction in both die area and power dissipation.

REFERENCE

[1] S.G. Mallat, “A Theory for Multiresolution Signal Decomposition: The Wavelet Representation”, IEEE Trans. on Pattern Analysis on Machine Intelligence, 110. July1989, pp. 674-693.

[2] X. Wu, Y. Li, and H. Chen, “programmable wavelet packet transform process,” IEEE Electronics Letters, vol. 35. no. 6, pp.449-450. 1999.

[3] K.K.Parhi, VLSI Digital Signal Processing. New York: Wiley, 1998.

[4] O. Farooq and S. Datta, “Mel filter-like admissible wavelet packet structure for speech recognition,” IEEE Signal Process. Lett. vol. 8, no.7, pp. 196-198, jul.2001.

[5] M. A. Trenas, J. Lopez and E. L. Zapata, “Architecture for wavelet packet transform”, J. VLSI Signal Process. Vol. 32. pp. 255-273, 2002.

[6] R. Mukesh and A,P. Vinod, “A new common sub-expression algorithm for implementing low-complexity FIR filter in software defined radio receivers” , in processing of IEEE International Symposium on Circuit and System, vol.4, pp.4515-4518, May 21-24 , 2006, Greece.

[7] M. A. Farahani, and M. Eshghi, “Architecture of a Wavelet Packet transform Using Parallel Filters” IEEE Trans. Signal Process, 36, 961-1005, 2006.

[8] Krishna Raj and Vedvrat, “A Study of VLSI Architectures for 2-D Discrete Wavelet Transform” 978-1-4244-8542-0/10/$26.00 ©2010 IEEE.

[9] Jitendra Jain, SaketShrivastava and SnehalataSahu, “Efficient VLSI Architecture for2-D Discrete WaveletTransform by using Xilinx Simulation” International Journal of Electronics and Computer Science Engineering ISSN 2277-1956/V2N2-582-588 (2012).

[10] SatyendraTripathi, Jitendra Jain, ArpitRawat, “Efficient VLSI Architecture for 2's Complement Based 2-D Discrete Wavelet Transform” International Journal of Electronics and Computer Science Engineering ISSN 2277-1956/V2N2-582-588 (2012).