International Journal on Applications in Information and Communication Engineering
Volume 1: Issue 6: June 2015, pp 12-16. www.aetsjournal.com ISSN (Online) : 2394-6237

High Throughput Multistandared Transform Core Implementation Based On Common Sharing Distributed Arithmetic

P.Malathy, V.Filomin Joseena

Abstract— This paper presents a low cost and high efficient multistandard transform supported video codec using combination of Factor Sharing and Distributed Arithmetic called Common Sharing distributed Arithmetic (CSDA) method. Common Sharing Distributed Arithmetic (CSDA) unit is used for provide MST. It is also used for provide the achievable accuracy. This architecture efficiently shares the available hardware resources, so the hardware cost gets reduced. It will support all video standards like MPEG1/2/4,VC-1,H.264 according to the selection of MUX. Simulation of MST core is done by using Model Sim.

Keywords- Common sharing distributed arithmetic (CSDA), discrete cosine transform (DCT), integer transform, multistandard transform (MST).

I. INTRODUCTION

Currently, several video compression standards, e.g., MPEG-2, MPEG-4, H.264/AVC and VC1 (Windows Media Video 9), are widely applied in video codec products, such as digital TV, mobile video, video conference, and so on. The intercommunications between the video devices using different standards are so much inconvenient, thus video codec supporting multiple standards are more useful and more attractive.

A codec is a device or software that is used to compress or decompress a digital media file, such as a video or song. The "codec" can be dividing into 2 parts: encode and decode. The encoder performs the compression (encoding) function and the decoder performs the decompression (decoding) function. Some codecs include both of these components and some codecs only include one of them.

A. Some Codecs

MPEG (Motion Picture Experts Group) is one of the biggest families in video codec, and it is the most common video format. The MPG, MPE, MPA, M15, M1V, MP2 etc. are all from this family. MPEG format, including MPEG video, MPEG audio and MPEG (video, audio synchronization) of three parts, MP3 (MPEG-3) audio files is a typical application of the MPEG audio, video include MPEG-1, MPEG-2 and MPEG4.

P.Malathy PG Student, M.E VLSI Design, Kings college of Engineering, Anna university, Thanjavur, Tamilnadu

V.Filomin Joseena , Assistant Professor , Kings college of Engineering, Anna university, Thanjavur, Tamilnadu

MPEG-1, its compression algorithm is widely used in the production of VCD and the download of some video clip. Almost all VCD is compressed using the Mpeg-1 format (*. Dat file format). Using MPEG-1 compression algorithm, a 120-minute film (the original video files) can be compressed to about the size of 1.2 GB, and the file format is generally mpg and dat files.

MPEG-2, its compression algorithm used in the production of the DVD (* vob - file formats), and also in some of the HDTV (high definition television) and high demand video editing, processing of the application. Using MPEG-2 compression algorithm could produce a 120-minute film (the original video files) in the size of about 4GB to 8GB, of course, image quality indicators are MPEG-1 can not be compared. Use this compressed file format made by the algorithm is generally the vob file.

MPEG-4 is a new compression algorithm, the use of this compression algorithm can be a 120 minute film (the original video files) is compressed to about 300MB. Now, this compression algorithm of the MPEG are used by many encoding formats, such as ASF, DivX, Xvid, mp4 (Apple, mpeg-4 encoding format) are using the MPEG-4 compression algorithm.

VC-1 is a video codec specification that has been standardized by the Society of Motion Picture and Television Engineers (SMPTE) and implemented by Microsoft as Microsoft Windows Media Video (WMV) 9.

B. Transform Techniques

The discrete cosine transform (DCT) is a technique for converting a signal into elementary frequency components. It is widely used in image compression. Here we develop some simple functions to compute the DCT and to compress images.

Distributed Arithmetic (DA) is a technique that is bit serial in nature. It can therefore appear to be slow It turns out that when the number of elements in a vector is nearly the same as the word size, DA is quite fast „ DA `replaces' the explicit multiplications by ROM look-ups. It is an efficient technique to implement on Field Programmable Gate Arrays (FPGAs).

The circuit area can be reduced efficiently by combining Distributed Arithmetic [1-8] and Factor Sharing [9-14] called as CSDA.Factor Sharing method shares the same factor in different coefficients among the same input in order to reduce cost.

II. MATHEMATICAL DERIVATION OF THE PROPOSED CSDA ALGORITHM

To gain better resource sharing for inner-product operation, the proposed CSDA combines the FS and DA methods. The foundations of the FS and DA method are described in the following.

A. Mathematical Derivation

The FS method shares the same factor in different coefficients among the same input. Consider two different elements S_1 and S_2 with the same input X as an example

$$S1=C1X,\ S2=C2X. \quad (1)$$

Assuming that the same factor Fs can be found in the coefficients C_1 and C_2, (1) can be rewritten as follows:

$$S_1 = (Fs2^{k1} + Fd1)\ X$$
$$S_2 = (Fs2^{k2} + Fd2)\ X \quad (2)$$

Where k1 and k2 indicate the weight position of the shared factor *Fs* in C1 and C2, respectively. Fd1 and Fd2 denote the remainder coefficients after extracting the shared factor Fs for C1 and C2, respectively

$$Fd1 = C1 - Fs2^{k1}$$
$$Fd2 = C2 - Fs2^{k2} \quad (3)$$

B. Mathematical Derivation of CSD Format Distributed Arithmetic

The inner product for a general matrix multiplication-and accumulation can be written as follows:

$$Y=A^TX=\sum_{i=1}^{L} A_iX_i \quad (4)$$

where *Ai* is an *N*-bit CSD coefficient, and *Xi* is the input data. Equation (4) can be expressed as listed below

$$Y=[2^0 \quad 2^{-1} \ldots \quad 2^{-(N-1)}]\times \begin{bmatrix} A_{1,0} & A_{2,0} \cdots & A_{L,0} \\ A_{1,1} & A_{2,1} & A_{L,1} \\ . & . & . \\ . & . & . \\ A_{1,N-1} & A_{2,-(N-1)} & A_{L,N-1} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ . \\ . \\ X_L \end{bmatrix}$$

$$= [2^0 \ 2^{-1} \ldots..2^{-(N-1)}] \begin{bmatrix} Y_0 \\ \vdots \\ Y_{N-1} \end{bmatrix} \quad (5)$$

C. Proposed CSDA Algorithm

The proposed CSDA algorithm combines the FS and DA methods. By expanding the coefficients matrix at the bit level, the FS method first shares the same factor in each coefficient; the DA method is then applied to share the same combination of the input among each coefficient position. An example of the proposed CSDA algorithm in a matrix inner product is as follows:

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \quad (6)$$

where the coefficients *C*11 ~ *C*22 are all five-bit CSD numbers

$C_{11}=[1 \quad -1 \quad 1 \quad 0 \quad 0]$
$C_{12}=[1 \quad -1 \quad 0 \quad 0 \quad 1]$
$C_{21}=[1 \quad 1 \quad -1 \quad 0 \quad 0]$
$C_{22}=[0 \quad 1 \quad -1 \quad 0 \quad 0]$ (7)

III. PROPOSED 2-D CSDA MST CORE DESIGN

A Block diagram of 2-D CSDA MST core design

For increasing the throughput and reduce the hardware cost, the 2-D CSDA (combines factor sharing and DA) core design has often been implemented using either **direct or indirect methods**. The direct methods include fast algorithms that reduce the computation complexity. On the other hand, the 2-D CSDA cores using the indirect method are implemented based on transpose memory and have the following two structures:

1. Two 1-D CSDA cores and one transpose memory
2. a single 1-D CSDA core and one TMEM.

In the first structure as shown in Fig.1, the 2-D CSDA core has a high throughput rate, Because the two 1-D DCT cores compute the transformation simultaneously.

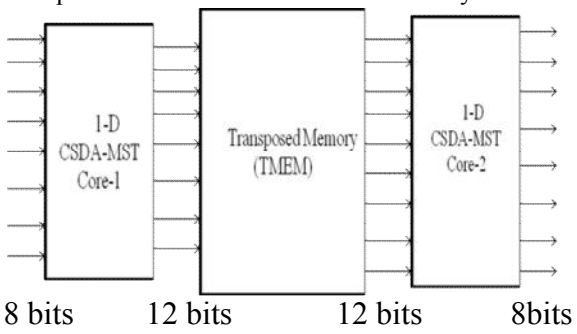


Fig.1 Block diagram of 2-D CSDA MST core

B Architecture of 1-D CSDA MST

In this section architecture of 1-D CSDA MST core as shown in Fig. 2 is discussed. This architecture consist of following modules:

Selected butterfly Module(SBF)
Even part CSDA
Odd part CSDA
Error Correction Adder Tree(ECAT)
Permutation

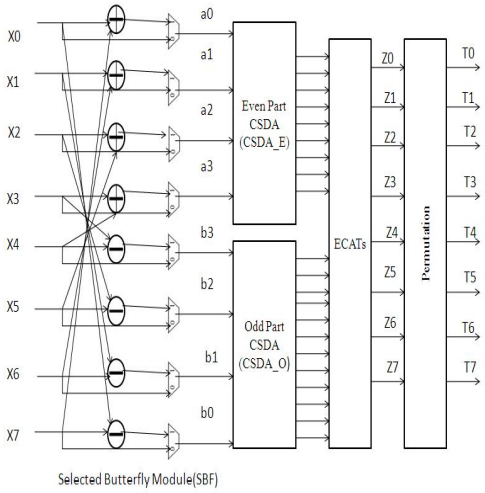


Fig.2 Architecure of 1-D CSDA MST

A. SBF

Selected butterfly module uses 8 multiplexers for modeling the 8 point butterfly inputs. These 8 point transforms are divided into two 4 point transform according to symmetry property called as even part CSDA and odd part CSDA.

B. EVEN PART CSDA

Even part (Fig.3) of the 8 point transform calculated by the CSDA even part module. It has two stages each stage have separate pipeline register architectures. First stage stage-executes four input butterfly matrix circuit. Second stage-it further decompose into odd and even part and by using CSDA algorithm to share the hardware resources in variable standards.

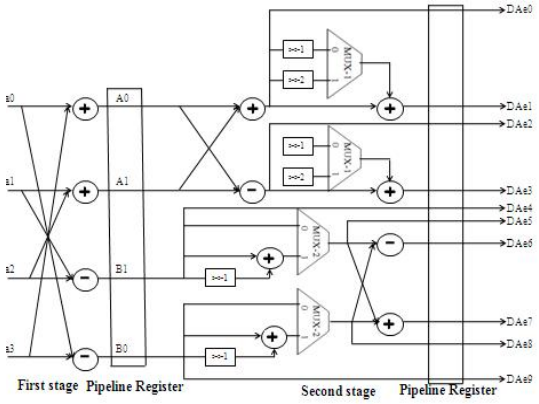


Fig .3 Architecture of even part CSDA

C. ODD PART CSDA

Similar to CSDA_E ,the CSDA_O(Fig.4) also share the hardware resources.

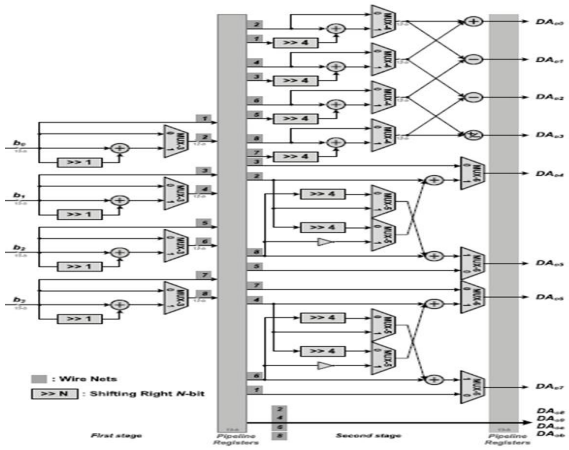


Fig .4

D. ECAT&Permutation

Eight adder trees with error compensation(ECATs) are followed by CSDA_E and CSDA_O, which add the nonzero CSDA coefficients with corresponding weights.

The ECATs circuits can alleviate truncation error efficiently in small area design when summing the nonzero data all together. Therefore, the hardware size and cost is reduced. Permutation module rearranging the into some sequence.

E. Transpose Memory

The architectural setup of the proposed transpose memory subsystem is shown in Fig. 5. It consists of an 8X8 array of 12 bit cell modules. The array is configured to receive inputs from the horizontal as well as vertical direction as well as to shift data in both horizontal as well as vertical directions.

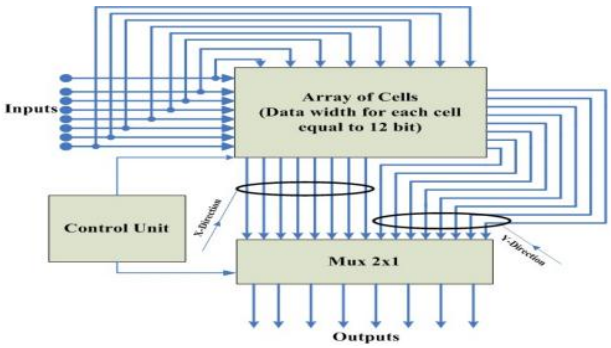


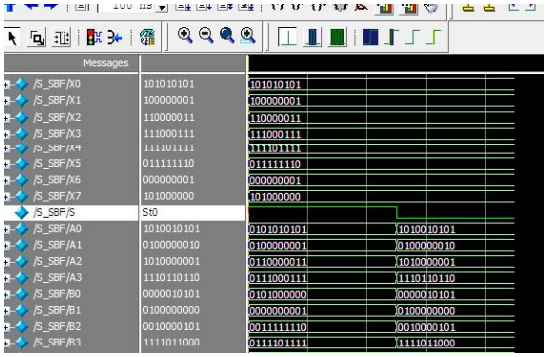
Fig. 5 Architecture of Transpose memory

One of the most important aspects of the proposed memory design is the ability to shift data in either direction as well as to switch the direction of data flow during the circuit operation. Each cell consists of a 12 bit register, 12 bit 2:1 multiplexer at the input and a 12 bit 2:1 de-multiplexer at the output. The multiplexer and de-multiplexer allow the register file to accept data from either direction as well as to transmit data in either horizontal or vertical direction. All the switching circuits,' i.e at the input as well as at the output, are controlled through a common control signal. The signal generated by the control unit switches the select lines of the multiplexer and de-multiplexer to switch the direction of data flow after every eight cycles. This implies that the array can now accept an 8X8 data block in the horizontal direction, and start transmitting the transpose through the vertical direction.

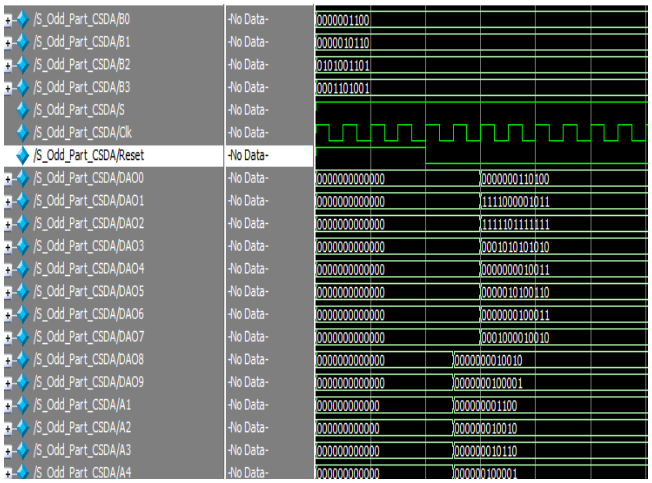
IV. SIMULATION RESULTS

The language which is used to build this architecture is verilog HDL. The simulation of this MST core is done by using Model Sim.

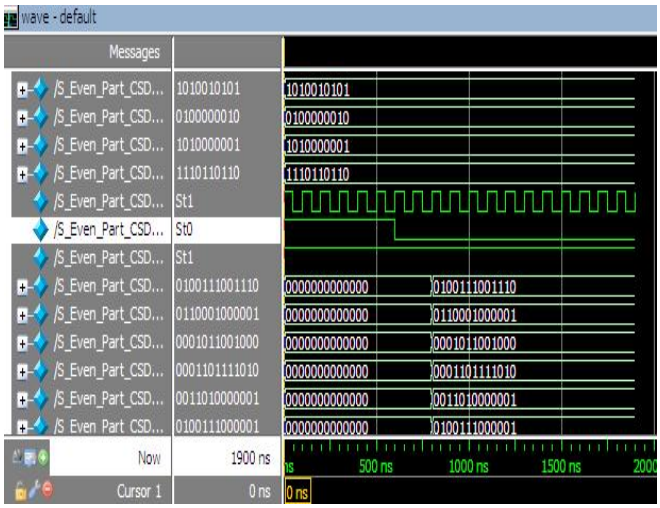
A. Simulation of SBF module



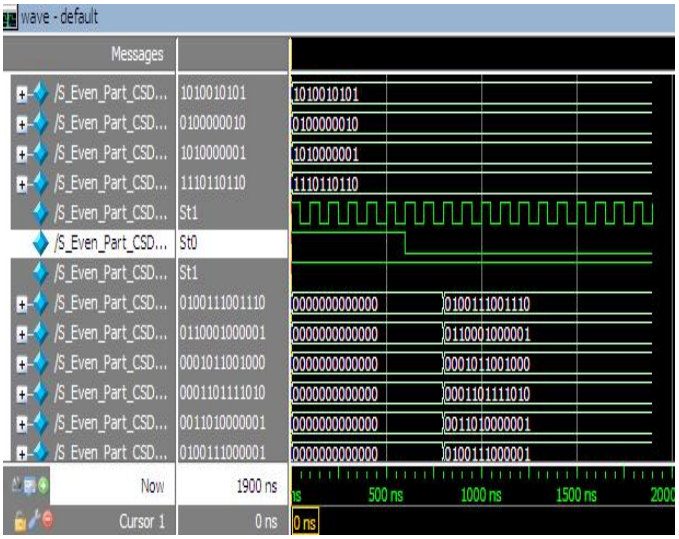
B. Simulation of odd part CSDA



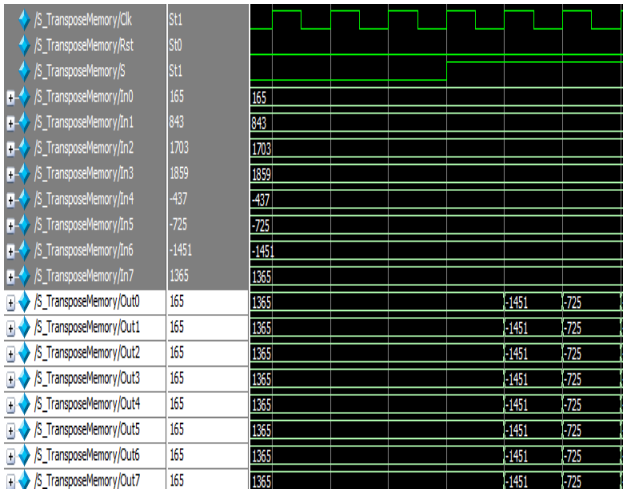
C. Simulation of even part CSDA



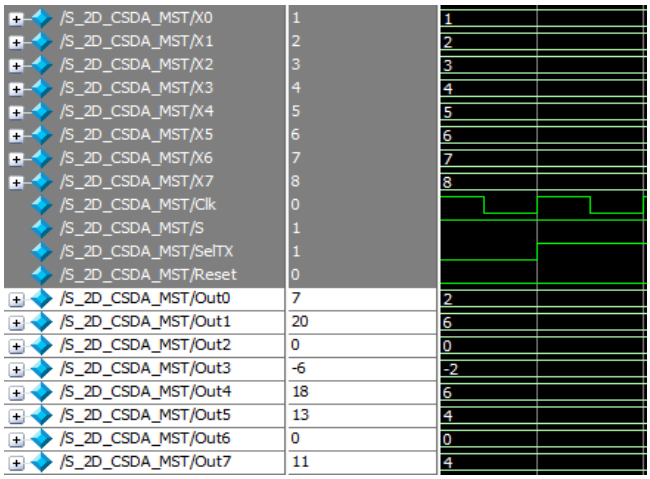
D. Simulation of 1-d CSDA



E. Transpose memory



F. 2-D CSDA MST core



G. Synthesis result of 2-D CSDA MST

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	1,259	9,312	13%
Number of 4 input LUTs	2,097	9,312	22%
Logic Distribution			
Number of occupied Slices	1,312	4,656	28%
Number of Slices containing only related logic	1,312	1,312	100%
Number of Slices containing unrelated logic	0	1,312	0%
Total Number of 4 input LUTs	2,277	9,312	24%
Number used as logic	2,097		
Number used as a route-thru	180		
Number of bonded IOBs	194	232	83%
Number of GCLKs	1	24	4%
Total equivalent gate count for design	28,624		
Additional JTAG gate count for IOBs	9,312		

The Output for 2-D Common Sharing Distributed Arithmetic- MST architecture gate count is 28.6 k. Compared with existing system ,it will support all video standards and also gate count is reduced.

V. CONCLUSION

The proposed CSDA MST core can support all video standards like MPEG-1/2/4,H.264 and VC-1 standards with high performance and low hardware resource cost. Common Sharing Distributed Arithmetic method efficiently combines both Factor Sharing and Distributed Arithmetic which reduces the gate counts. CSDA-MST core with a throughput rate , which can support (4928 × 2048@24 Hz) digital cinema format with only 30k logic gates.

ACKNOWLEDGMENT

Apart from our efforts, the success of any work depends on the support and guidelines of others. We take this opportunity to express our gratitude to the people who have been supported us in the successful completion of this work. We owe a sincere prayer to the LORD ALMIGHTY for his kind blessings without which this would not have been possible.

REFERENCES

[1] S. Uramoto, Y. Inoue, A. Takabatake, J. Takeda, Y. Yamashita, H. Terane, and M. Yoshimoto, "A 100-MHz 2-D discrete cosine transform core processor," *IEEE J. Solid-State Circuits*, vol. 27, no. 4, pp. 492–499,Apr. 1992.

[2] S. Yu and E. E. Swartzlander, "DCT implementation with distributed arithmetic," *IEEE Trans. Comput.*, vol. 50, no. 9, pp. 985–991, Sep. 2001.

[3] A. M. Shams, A. Chidanandan, W. Pan, and M. A. Bayoumi, "NEDA: A low-power high-performance DCT architecture," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 955–964, Mar. 2006.

[4] C. Peng, X. Cao, D. Yu, and X. Zhang, "A 250 MHz optimized distributed architecture of 2D 8×8 DCT," in *Proc. 7th Int. Conf. ASIC*, Oct. 2007, pp. 189–192.

[5] C. Y. Huang, L. F. Chen, and Y. K. Lai, "A high-speed 2-D transform architecture with unique kernel for multi-standard video applications," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2008, pp. 21–24.

[6] Y. H. Chen, T. Y. Chang, and C. Y. Li, "High throughput DA-based DCT with high accuracy error-compensated adder tree," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 4, pp. 709–714, Apr. 2011.

[7] Y. H. Chen, T. Y. Chang, and C. Y. Li, "A high performance video transform engine by using space-time scheduling strategy," *IEEE Trans.Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 4, pp. 655–664, Apr. 2012.

[8] Y. K. Lai and Y. F. Lai, "A reconfigurable IDCT architecture for universal video decoders," *IEEE Trans. Consum. Electron.*, vol. 56, no. 3, pp. 1872–1879, Aug. 2010.

[9] H. Chang, S. Kim, S. Lee, and K. Cho, "Design of area-efficient unified transform circuit for multi-standard video decoder," in *Proc. IEEE Int.SoC Design Conf.*, Nov. 2009, pp. 369–372.

[10] S. Lee and K. Cho, "Circuit implementation for transform and quantization operations of H.264/MPEG-4/VC-1 video decoder," in *Proc. Int. Conf. Design Technol. Integr. Syst. Nanosc.*, Sep. 2007, pp. 102–107.

[11] S. Lee and K. Cho, "Architecture of transform circuit for video decoder supporting multiple standards," *Electron. Lett.*, vol. 44, no. 4, pp. 274–275, Feb. 2008.

[12] H. Qi, Q. Huang, and W. Gao, "A low-cost very large scale integration architecture for multistandard inverse transform," *IEEE Trans. Circuits Syst.*, vol. 57, no. 7, pp. 551–555, Jul. 2010.

[13] T. S. Chang, C. S. Kung, and C. W. Jen, "A simple processor core design for DCT/IDCT," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10,no. 3, pp. 439–447, Apr. 2000.

[14] K. H. Chen, J. I. Guo, J. S. Wang, C. W. Yeh, and T. F. Chen, "A power aware IP core design for the variable-length DCT/IDCT targeting at MPEG4 shape-adaptive transforms," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 2. May 2004, pp. 141–144.

[15] S. Lee and K. Cho, "Design of high-performance transform and quantization circuit for unified video CODEC," in *Proc. IEEE Asia Pacific Conf. Circuits Syst.*, Nov. 2008, pp. 1450–1453.

[16] C. P. Fan and G. A. Su, "Fast algorithm and low-cost hardware-sharing design of multiple integer transforms for VC-1," *IEEE Trans. Circuits Syst.*, vol. 56, no. 10, pp. 788–792, Oct. 2009.

[17] C. P. Fan and G. A. Su, "Efficient fast 1-D 8×8 inverse integer transform for VC-1 application," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 4, pp. 584–590, Apr. 2009.

[18] F. Xu, C. H. Chang, and C. C. Jong, "Design of low-complexity FIR filters based on signed-powers-of-two coefficients with reusable common subexpressions," *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.*, vol. 26, no. 10, pp. 1898–1907, Oct. 2007.

[19] Y. J. Yu and Y. C. Lim, "Optimization of FIR filters in subexpression space with constrained adder depth," in *Proc. IEEE 6th Int. Symp. Image Signal Process. Anal.*, Sep. 2009, pp. 766–769.

[20] M. Martina and G. Masera, "Multiplierless, folded 9/7–5/3 wavelet VLSI architecture," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 9,pp. 770–774, Sep. 2007.

[21] Z. Yu, M. J.Meeuwsen, R. W. Apperson, O. Sattari, M. Lai, J. W.Webb, E. W. Work, D. Truong, T. Mohsenin, and B. M. Baas, "AsAP: An asynchronous array of simple processors," *IEEE J. Solid-State Circuits*, vol. 43, no. 3, pp. 695–705, Mar. 2008.