ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Implementation of Multi Standard Transform For High Throughput Core Supporting Mpeg/H.264/Vc-1 Based on CSDA

T.Shivalingappa
PG Scholar,
Department of ECE,
Intellectual Institute of Technology, AP, India.

K.Madan Mohan
Assistant Professor,
Department of ECE,
Intellectual Institute of Technology, AP, India.

Abstract:

Transform are widely used in video and image applications. proposes a low-cost high-throughput Multi standard transform (MST) core, which can support MPEG- 1/2/4 (8 × 8), H.264 (8 × 8, 4 × 4), and VC-1 (8 × 8, 8 × 4, 4×8, 4×4) transforms. Common sharing distributed arithmetic (CSDA) combines factor sharing and distributed arithmetic sharing techniques, efficiently reducing the number of adders for high hardware-sharing capability. With eight parallel computation paths, the proposed MST core has an eightfold operation frequency throughput rate. Measurements show that the proposed CSDAMST core achieves a high-throughput rate the proposed CSDA combines the FS and DA methods. The FS method is adopted first to identify the factors that can achieve higher capability in hardware resource sharing. The DA method is used to find the shared coefficient based on the results of the FS method. The adder-tree circuits will be followed by the proposed CSDA circuit.

Index Terms:

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

1 INTRODUCTION:

Transforms are widely used in video and image applications. Several groups, such as The International Organization for Standardization (ISO), International Telecommunication Union Telecommunication Standardization Sector (ITU-T), and Microsoft Corporation, have developed various transform dimensions and coefficients, corresponding to different applicatios . Numerous researchers have worked on transform core designs, including discrete cosine transform (DCT) and integer transform, using distributed arithmetic (DA) factor sharing (FS) and matrix decomposition methods to reduce hardware cost.

The inner product can be implemented using ROMs and accumulators instead of multipliers to reduce the area cost . Yu and Swartzlander present an efficient method for reducing ROMs size with recursive DCT algorithms. Although ROMs are likely to scale much better than other circuits with shrinking technology nodes, several ROM-free DA architectures have recently emerged. Shams et al. employ a bit-level sharing scheme to construct the adder-based butterfly matrix, called new DA (NEDA). To improve the throughput rate of the NEDA method, high-throughput adder trees are introduced in.

Chang et aluse a delta matrix to share hardware resources using the FS method. They derive matrices for multi standards as linear combinations from the same matrix and delta matrix, and show that the coefficients in the same matrix can share the same hardware resources by factorization To further reduce the area, Qi et al. present optimization strategies for FS and adder sharing (AS) for multistandard (MST) applications. Fan and Su use the matrix decomposition method to establish the sharing circuit. Matrices for VC-1 transformations can be decomposed into several small matrices, a number of which are identical for different points transforms. Hardware resources can be shared.

Moreover, other pervious works on hardware resource sharing are presented. Recently, reconfigurable architectures have been presented as a solution to achieve a good flexibility of processors in field-programmable gate array (FPGA) platform or application-specific integrated circuit (ASIC), such as AsAP, Ambric , MORA , and Smart Cell . Although these reconfigurable architectures have the feature of flexibility, the pure ASIC design can be recommended for a fixed customer application suitably. Because of the same properties in DCT and integer transform applied to Moving Picture Experts Group (MPEG) and Windows Media Video 9 (WMV-9/VC-1) , many MST cores are presented Hwangbo and Kyung introduce a fully supported transform core for the H.264 standard, including 8 × 8 and 4 × 4 transforms.

2COMMON SHARING DISTRIBUTED ALGORITHM:

Mathematical derivation of proposed Common sharing distributed Algorithm.To gain better resorce sharing of inner product operation, The proposed CSDA method combines Factor Sharing and Distributed methods. To find FS and DA methods are described in the following.

A.Mathematical derssivation of Factor Sharing:

The factor sharing method shares the same factor in different coefficients of applied input .consider two different elements s1 and s2 with same input X as an example.
S1=C1X , S2=C2X
Assuming the same factor Fs can befound in coefficients C1 and C2 , S1 and S2 can be rewritten as follows.
S1=(Fs2k1+Fd1)X
S2=(Fs2k2+Fd2)X
Where k1 and k2 indicates the weight position of the sharing factor Fs in C1 and C2 respectively.Fd1 and Fd2 denote the remainder coefficients after extracting sharing factor Fs for
C1 and C2 respectively.
Fd1 = C1-Fs2k1
Fd2 = C2-Fs2k2

b.Mathematical derivation of Common sharing distributed arithmetic:

The inner product for a general multiplication and accumulation can be written as

$$Y=A^T X=\sum_{i=1}^{L} A_i X_i$$

where Xi is input data,Ai is N-bit CSD co efficientit can be expressed as follow.

$$Y=[2^0\ 2^{-1}\ \cdots\ 2^{-(N-1)}] \cdot \begin{bmatrix} A_{1,0} & A_{2,0} & \cdots & A_{L,0} \\ A_{1,1} & A_{2,1} & \cdots & A_{L,1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{1,(N-1)} & A_{2,(N-1)} & \cdots & A_{L,(N-1)} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_L \end{bmatrix}$$

$$= [2^0\ 2^{-1}\ \cdots\ 2^{-(N-1)}] \begin{bmatrix} Y_0 \\ Y_1 \\ \vdots \\ Y_{(N-1)} \end{bmatrix}$$

Where $Y_i=\sum_{i=1}^{L} A_i X_i , A_{i,j} \in \{-1\ ,0,1\}$ and

j=0,…,(N-1).If Yj can be calculated by adding or subtrscting Xi with Ai,j ≠ 0. The product Y can then be obtained by shifting and adding everynon zero Yj.

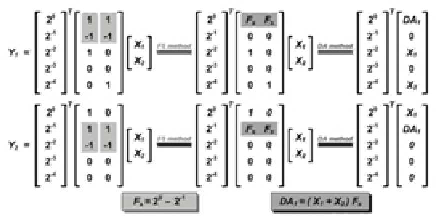


Fig. 1. Example of a CSDA algorithm.

C.1-D Common Sharing Distributed arithmetic-MST:

Based on the proposed CSDA algorithm, the coefficients for MPEG-1/2/4, H.264, and VC-1 transforms are chosen to achieve high sharing capability for arithmetic resources. To adopt the searching flow, software code will help to do the iterative searching loops by setting a constraint with minimum nonzero elements. in this paper, the constraint of minimum nonzero elements is set to be five. After software searching, the coefficients of the CSD expression, where 1 indicates −1. Note that the choice of shared coefficient is obtained by some constraints. Thus, the chosen CSDA coefficient is not a global optimal solution. It is just a local or suboptimal solution. Besides, the CSD codes are not the optimal expression, which have minimal nonzero bits. However, the chosen coefficient of CSD expression can achieve high sharing capability for arithmetic resources by using the proposed CSDA algorithm. More information about CSDA coefficients for MPEG-1/2/4, H.264, and VC-1 transforms.

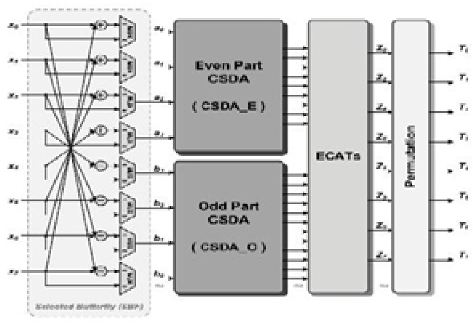


Fig.1.Architecture of proposed 1-D CSDA-MST.

Even part common sharing distributed arithmetic circuit:

The SBF module executes for the eight-point transform and bypasses the input data for two four-point transforms.

After the SBF module, the CSDA_E and CSDA_O execute and by feeding input data a and b, respectively. The CSDA_E calculates the even part of the eight-point transform, similar to the four-point Trans form for H.264 and VC-1 standards.

Within the architecture of CSDA_E, two pipeline stages exist (12-bit and 13-bit). The first stage executes as a four-input butterfly matrix circuit, and the second stage of CSDA_E then executes by using the proposed CSDA algorithm to share hardware resources in variable standards.

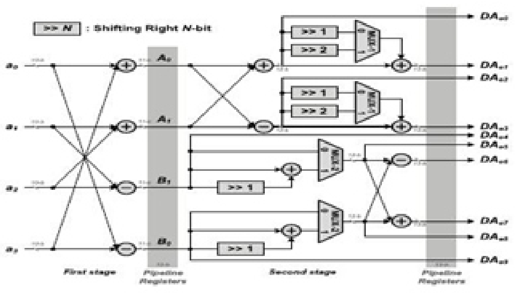


Fig.2 Architectre of even part CSDA

Odd part common sharing distributed arithmeti circuit:

Similar to the CSDA_E, the CSDA_O also has two pipeline stages. Based on the proposed CSDA algorithm, the CSDA_O efficiently shares the hardware resources among the odd part of the eight-point transform and four-point transform for variable standards.

It contains selection signals of multiplexers (MUXs) for different standards. Eight adder trees with error compensation (ECATs) are followed by the CSDA_E and CSDA_O, which add the nonzero CSDA coefficients with corresponding weight as the tree-like architectures. The ECATs circuits can alleviate truncation error efficiently in small area design when summing the nonzero data all together.

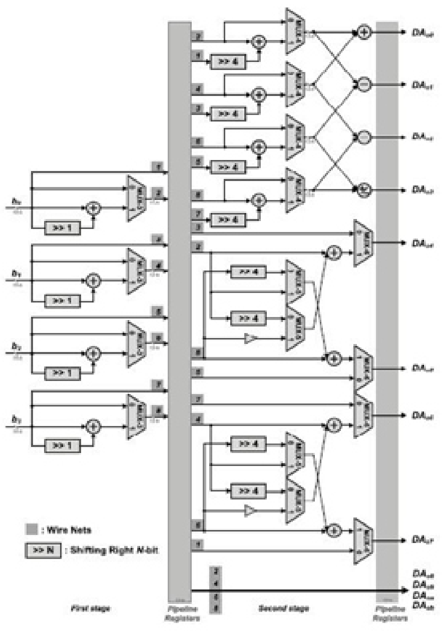


Fig.3 Architectre of odd part CSDA

d. Proposed CSDA Algorithm:

the proposed CSDA combines DA and FS methods. By expand the coefficients matrix at bit level The Factor sharing method first shares the same factor in each coefficient ,the distributed method is then applied to share the same combination of Input among each coefficient position. The proposed CSDA algorithm in matrix inner product can explain as follows.

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

Where thecoefficients C11 ~ C22 are all five bits CSD numbers
C11=[1 -1 1 0 0]
C12=[1 -1 0 0 1]
C21=[1 1 -1 0 0]
C22=[0 1 -1 0 0] (6)

Fig.5. shows the proposed CSDA sharing flow.The shared factor Fs in four coefficients is [1 -1] and C11 ~ C22 can use instead of [1 -1] ,with the the corresponding position under the FS method. The Distributed Arithmetic is applied to share the same position for the input, and the DA shared coefficient DA1=(X1+X2)Fs. Finally ,the matrix inner product in above equation can be implemented by shifting and adding every nonzero weight position. Fig, Shows the coefficient searching flow of the proposed Common Sharing Distributed Arithmetic algorithm.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

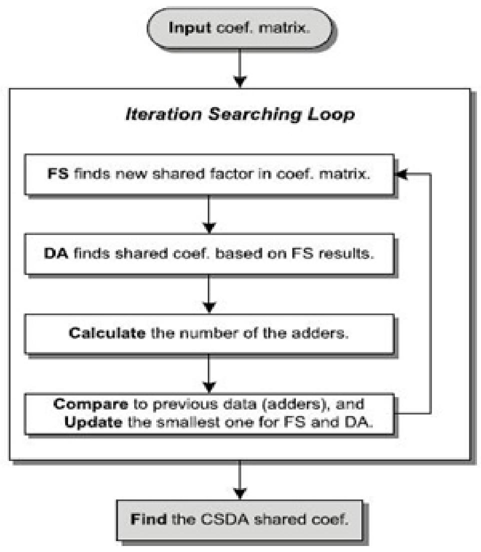

Input coef. matrix.
Iteration Searching Loop
FS finds new shared factor in coef. matrix.
DA finds shared coef. based on FS results.
Calculate the number of the adders.
Compare to previous data (adders), and Update the smallest one for FS and DA.
Find the CSDA shared coef.

Fig.5. The Proposed CSDA searching Flow.

2-D common sharing distributed arithmetic-MST core: This section provides a discussion of the hardware resources and system accuracy for the proposed 2-D CSDA-MST core and also presents a comparison with previous works. Finally, the characteristics of the implementation into a chip are described.

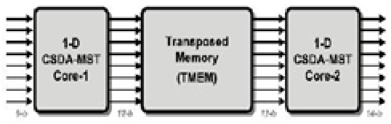
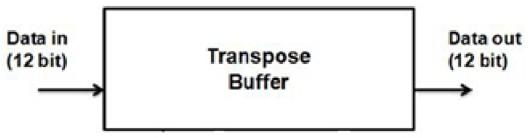


Fig.6.Proposed 2-D CSDA-MST core.

TMEM:

The TMEM is implemented using 64-word 12-bit dual-port registers and has a latency of 52 cycles. Based on the time scheduling strategy and result of the time scheduling strategy, the 1st-D and 2nd-D transforms are able to be computed simultaneously. The transposition memory is an 8×8 register array with the data width of 16 bits and is shown in Fig.7.



a. Mathematical derivation of eight-point and four-point transrms:

We introduces the proposed 2-D CSDA-MST core implementation. Neglecting the scaling factor, the one dimensional (1-D) eight-point transform can be defined as follows.

$$\begin{bmatrix} Z_0 \\ Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \\ Z_6 \\ Z_7 \end{bmatrix} = \mathbf{C} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} c_4 & c_4 & c_4 & c_4 & c_4 & c_4 & c_4 & c_4 \\ c_1 & c_3 & c_5 & c_7 & -c_7 & -c_5 & -c_3 & -c_1 \\ c_2 & c_6 & -c_6 & -c_2 & -c_2 & -c_6 & c_6 & c_2 \\ c_3 & -c_7 & -c_1 & -c_5 & c_5 & c_1 & c_7 & -c_3 \\ c_4 & -c_4 & -c_4 & c_4 & c_4 & -c_4 & -c_4 & c_4 \\ c_5 & -c_1 & c_7 & c_3 & -c_3 & -c_7 & c_1 & -c_5 \\ c_6 & -c_2 & c_2 & -c_6 & -c_6 & c_2 & -c_2 & c_6 \\ c_7 & -c_5 & c_3 & -c_1 & c_1 & -c_3 & c_5 & -c_7 \end{bmatrix}.$$

Because the eight-point coefficient structures in H.264, MPEG- 1/2/4, and VC-1 standards are the same, the eight-point transform for these standards can use the same mathematic derivation. According to the symmetry property, the 1-D eight point transform in (7) can be divided into even and odd two four-point transforms, Ze and Zo, as listed in (8) and(9),respectively.

$$\mathbf{Z}_e = \begin{bmatrix} Z_0 \\ Z_2 \\ Z_4 \\ Z_6 \end{bmatrix} = \begin{bmatrix} c_4 & c_4 & c_4 & c_4 \\ c_2 & c_6 & -c_6 & -c_2 \\ c_4 & -c_4 & -c_4 & c_4 \\ c_6 & -c_2 & c_2 & -c_6 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

$$=\mathrm{C_e.a} \qquad (8)$$

$$\mathbf{Z}_o = \begin{bmatrix} Z_1 \\ Z_3 \\ Z_5 \\ Z_7 \end{bmatrix} = \begin{bmatrix} c_1 & c_3 & c_5 & c_7 \\ c_3 & -c_7 & -c_1 & -c_5 \\ c_5 & -c_1 & c_7 & c_3 \\ c_7 & -c_5 & c_3 & -c_1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$=\mathrm{C_o.b} \qquad (9)$$

Where

$$\mathbf{a} = \begin{bmatrix} x_0 + x_7 \\ x_1 + x_6 \\ x_2 + x_5 \\ x_3 + x_4 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} x_0 - x_7 \\ x_1 - x_6 \\ x_2 - x_5 \\ x_3 - x_4 \end{bmatrix}. \qquad (10)$$

The even part of the operation in (9) is the same as that of the four-point VC-1 and H.264 transformations. Moreover, the even part Ze can be further decomposed into even and odd parts: Zee and Zeo

$$\mathbf{Z}_{ee} = \begin{bmatrix} Z_0 \\ Z_4 \end{bmatrix} = \begin{bmatrix} c_4 & c_4 \\ c_4 & -c_4 \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \end{bmatrix}$$

$$=\mathbf{C_{ee}.A}$$

$$\mathbf{Z}_{eo} = \begin{bmatrix} Z_2 \\ Z_6 \end{bmatrix} = \begin{bmatrix} c_2 & c_6 \\ c_6 & -c_2 \end{bmatrix} \begin{bmatrix} B_0 \\ B_1 \end{bmatrix}$$

$$=\mathbf{C_{eo}.B}$$

$$\mathbf{A} = \begin{bmatrix} a_0 + a_3 \\ a_1 + a_2 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} a_0 - a_3 \\ a_1 - a_2 \end{bmatrix}.$$

b. Architecture of the Proposed 2-D CSDA-MST Core:

The architecture of the 1-D eight-point MST core is shown in Fig.1 , which consists of a selected butterfly (SBF) module, an even part CSDA (CSDA_E),an odd part CSDA(CSDA_O),eight error-compensated error trees (ECATs), and a permutation module. The following example serves to clarify the operation of the proposed CSDA algorithm. When the MPEG-1/2/4 standard is conducted, Z0 can be obtained by executing c4 A0 +c4A1.
Z0 = c4A0 + c4A1
= (M1 2-2 +M2 2-5 + M1 2-7) A0
+ (M1 2 -2 + M2 2-7 +M1 2-7) A1
=Dee0 2-2
+ Dee1 2-5 + Dee0 2-7
=DAe12-2 + DAe02-5+DAe12-7
Where
DAe0=Dee1= (A0+A1) M2
DAe1=Dee0= (A0+A1) M1
The nonzero DA factors DA0 and DAe1 can subsequently be Conducted using CDSA_E and Z0 can be obtained in ECAT by summing DAe0 and DAe1 with the corresponding weights. Before the 1-D transform output, the permutation module re permutes the transform outputs Z to T between eight point and four-point transformations. As with the eight-point transform, the output data T are
[T0 T1 T2 …..T7] = [Z0 Z1 Z2………..Z7] (14)
Because the CSDA_O can execute the four-point transform In, the transform output data T for the four-point transform are
[T0 T1 T2 T3] = [Z0 Z2 Z4 Z6]
[T4 T5 T6 T7] = [Z1 Z3 Z5 Z7].
The SBF module executes (10) for the eight-point transform and bypasses the input data for two four-point transforms. After the SBF module, the CSDA_E and CSDA_O execute (9) and (9) by feeding input data a and b, respectively. The CSDA_E calculates the even part of the eight-point transform (9), similar to the four-point transform for H.264 and VC-1 Standards. Within the architecture of CSDA_E (Fig.2), two pipeline stages exist (12-bit and 13-bit). The first stage executes (13) as a four-input butterfly matrix circuit, and the second stage of CSDA_E then executes (11) and (12) by using the proposed CSDA algorithm to share hardware resources in variable standards. Similar to the CSDA_E, the CSDA_O also has two pipeline stages (Fig. 3). Based on the proposed CSDA algorithm, the CSDA_O efficiently shares the hardware resources among the odd part of the eight-point transform in (9) and four-point transform in (9) for variable standards.The proposed system by reducing the Area of converting one dimensional to two dimensional core designs. And also will reduce the total number of adders by using the CSDA Common Sharing Distributed Arithmetic method in the proposed system.

4. SIMULATION RESULTS:

The Multi standard transform core written in verilog, compiled and simulation using modelsim. The circuit simulated and synthesized. The simulated result for Multi standard transform core.

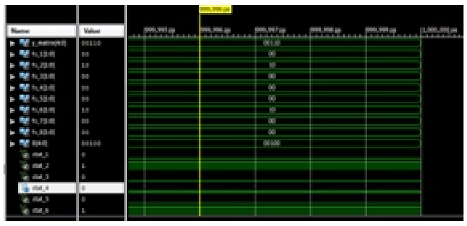


Fig. 5 Simulation Result.

10. CONCLUSION:

The CSDA-MST core can achieve high performance, with a high throughput rate and low-cost VLSI design, supporting MPEG-1/2/4, H.264, and VC-1 MSTs. By using the proposed CSDA method, the number of adders and MUXs in the MST core can be saved efficiently. Measured results show the CSDA-MST core with a throughput rate of 1.28 G-pels/s, which can support (4928 × 2048@24 Hz) digital cinema format with only 30 k logic gates. Because visual media technology has advanced rapidly, this approach will help meet the rising high-resolution specifications and future needs as well.

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 poweraware IP core design for the variable-length DCT/IDCT targeting atv 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.