High-Throughput Multistandard Transform Core Supporting H.264/MPEG/VC-1 using Common Sharing Distributed Arithmetic

K.Babu, P.Pavan Kumar Reddy, Panem charan Arur

Abstract— **This paper proposes low-cost improving high-throughput video transformations using multi standard transform (MST) core, and common sharing distributed algorithm. multi standard transform core can support, H.264 (8 × 8, 4 × 4), MPEG- 1/2/4 (8 × 8) and VC-1 (8 × 8, 8 × 4, 4×8, 4×4) transforms. Common sharing distributed arithmetic (CSDA) combines distributed arithmetic and factor sharing techniques, efficiently reducing the number of adders for high hardware-sharing capability. This achieves 44.5% reduction in adders in the proposed MST, compared with the direct implementation method. With eight parallel computation paths, the proposed MST core has an eightfold operation frequency throughput rate. The CSDA-MST core achieves a high-throughput rate supporting multi standard transformations at low cost and it measures to achieve a high-throughput rate of 1.28 G-pels/s, supporting the (4928×2048@24H) digital cinema or ultrahigh resolution formate.**

Index Terms-Common sharing distributed arithmatic (CSDA), discrete cosine trnasform(DCT), Integer transform, Multistandard trnsform (MST)

I.INTRODUCTION

TRANSFORMS are widely used in image and video Applications. Several groups, such as The International Telecommunication Union Telecommunication Standardization Sector (ITU-T), Microsoft Corporation and International Organization for Standardization (ISO) have developed various Transforms and coefficients, Corresponding to different applications .

Several researchers have worked on transform core designs, including integer Transform and discrete cosine transform, distributed arithmetic[1]-[2], factor sharing technique [1]-[3] and matrix decomposition methods to reducing hardware cost. This system include ROM Accumulators instead of multipliers to store reduce area cost. An efficient method for reducing ROMs size with recursive Discrete Cosine Transform(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. 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. It uses 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 present optimization strategies for Factor Sharing and adder sharing. For multi standard applications. It can also 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, such as AsAP. 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), H.264 and Windows Media Video 9 (WMV-9/VC-1), many MST cores are presented in. Introduces fully supported transform core for the H.264 standard, including 8 × 8 and 4 × 4 transforms. The eight-point and four-point transform cores for MPEG-1/2/4 and H.264 cannot support the VC-1 compression standard. In 8×8, 8×4, 4×8, and 4×4 transform cores are shown for VC-1, whereas MST cores supporting the MPEG-1/2/4, H.264, and VC-1 Standards are addressed.

This paper proposes a MST core that supports, H.264 (8 × 8, 4 × 4) MPEG-1/2/4 (8 × 8), and VC-1 (8 × 8, 8 × 4, 4 × 8, 4 × 4) transforms. The proposed MST core employs Distributed algorithm and Factor Sharing schemes as common sharing distributed arithmetic (CSDA) to reduce hardware cost. The main strategy aims to reduce the nonzero elements using CSDA algorithm; thus, few adders are needed in the adder-tree circuit. According to the proposed mapping strategy, the chosen canonic signed digital (CSD) coefficients can achieve excellent sharing capability for hardware resources. When implemented in a 1.8-V TSMC 0.18-μm CMOS process, the proposed CSDA-MST core has a throughput rate of 1.28 gig pixels/s (G-pels/s) at 160 MHz The rate of 1.28 G-pels/s meets the (4928 × 2048@24 Hz) digital cinema or ultrahigh resolution format.

H.264 and VC-1 are popular video compression standards. The VC-1 codec is designed to achieve state-of-the-art compressed video quality at bit rates that may range from very low to very high. The codec can easily handle 1920 pixel × 1080 pixel presentation at 6 to 30 megabits per second (Mbps) for high-definition video. VC-1 is capable of higher resolutions such as 2048 pixels × 1536 pixels for digital cinema, and of a maximum bit rate of 135 Mbps. An example of very low bit rate video would be 160 pixel × 120 pixel

presentation at 10 kilobits per second (Kbps) for modem applications. The basic functionality of VC-1 involves a block-based motion compensation and spatial transform scheme similar to that used in other video compression standards since MPEG-1 and H.261. However, VC-1 includes a number of innovations and optimizations that make it distinct from the basic compression scheme, resulting in excellent quality and efficiency. VC-1 Advanced Profile is also transport and container independent. This provides even greater flexibility for device manufacturers and content services. Traditionally, 8×8 transforms have been used for image and video coding. However, there is evidence to suggest that 4×4 transforms can reduce ringing artifacts at edges and discontinuities. VC-1 is capable of coding an 8 × 8 block using either an 8 × 8 transform, two 8 × 4 transforms, two 4 × 8 transforms, or four 4 × 4 transforms. This feature enables coding that takes advantage of the different transform sizes as needed for optimal image quality.

II.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 s_1 and s_2 with same input X as an example.

$$S_1=C_1X \text{ , } S_2=C_2X \qquad (1)$$

Assuming the same factor F_s can befound in coefficients C_1 and C_2, S_1 and S_2 can be rewritten as follows.

$$S_1=(F_s2^{k1}+F_{d1})X$$

$$S_2=(F_s2^{k2}+F_{d2})X \qquad (2)$$

Where k_1 and k_2 indicates the weight position of the sharing factor F_s in C_1 and C_2 respectively.F_{d1} and F_{d2} denote the remainder coefficients after extracting sharing factor F_s for C_1 and C_2 respectively.

$$Fd1 = C_1-F_s2^{k1}$$

$$Fd2 = C_2-F_s2^{k2} \qquad (3)$$

b.Mathematical derivation of Common sharing distributed arithmetic:

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

$$Y=A^T X = \sum_{i=1}^{L} A_iX_i \qquad (4)$$

where X_i is input data,A_i is N-bit CSD coefficientit can be expressed as follow.

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

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

Where $Y_i = \sum_{i=1}^{L} A_iX_i$,$A_{i,j} \in \{-1, 0, 1\}$ and j=0, …, (N−1). If Y_j can be calculated by adding or subtrscting X_i with $A_{i,j} \neq 0$. The product Y can then be obtained by shifting and adding everynon zero Y_j.

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 coefficients 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.

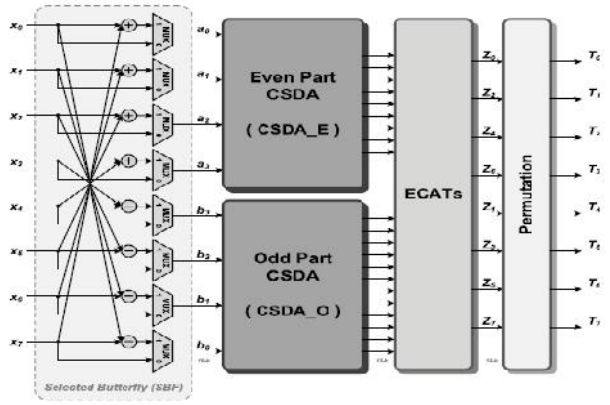


Fig.1.Architecture of proposed 1-D CSDA-MST.

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.

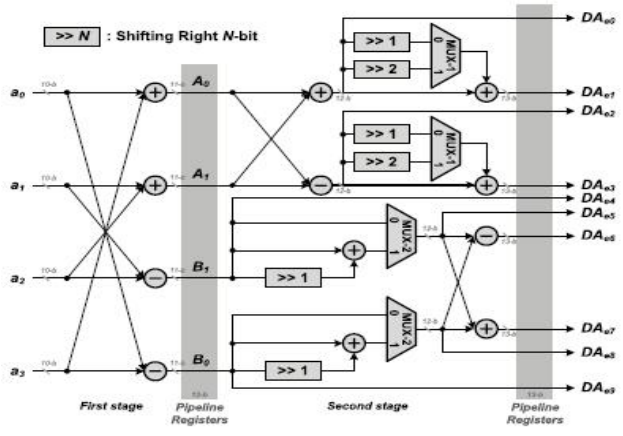


Fig.2 Architectre of even part CSDA

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.

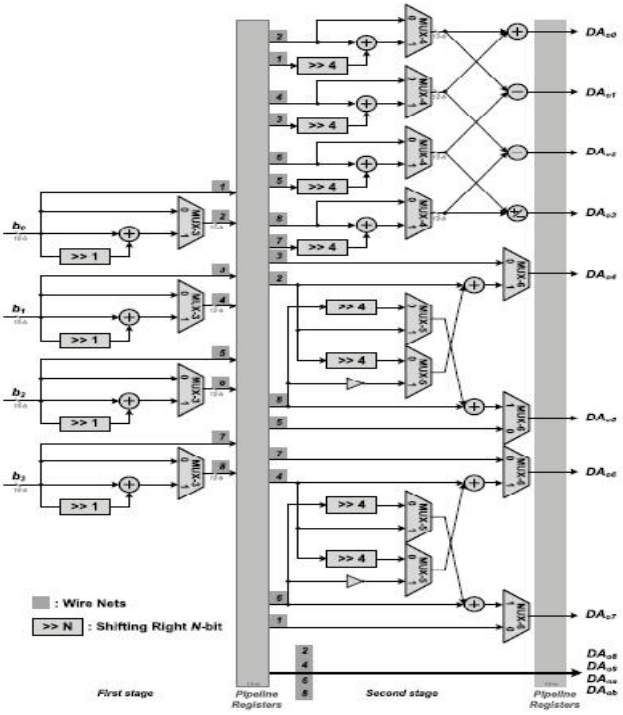


Fig.3 Architectre of odd part CSDA

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.

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \quad (5)$$

Where thecoefficients $C_{11} \sim C_{22}$ are all five bits CSD numbers

C_{11}=[1 -1 1 0 0]

C_{12}=[1 -1 0 0 1]

C_{21}=[1 1 -1 0 0]

$$C_{22}=[\ 0\ 1\ -1\ 0\ 0\ ] \quad (6)$$

Fig.5. shows the proposed CSDA sharing flow.The shared factor F_s in four coefficients is [1 -1] and $C_{11} \sim C_{22}$ 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 $DA_1=(X_1+X_2)F_s$. Finally ,the matrix inner product in above equation can be implemented by shifting and adding every nonzero weight position.

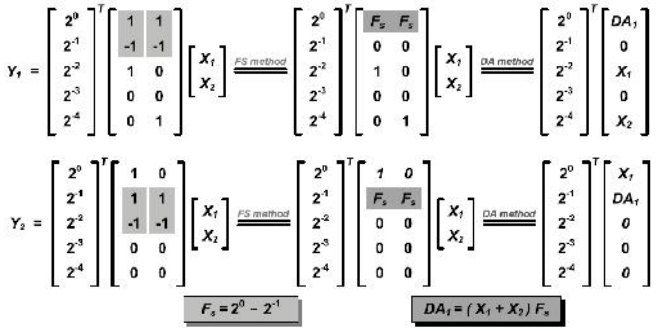


Fig.4. Example of CSDA Algorithm

Fig, Shows the coefficient searching flow of the proposed Common Sharing Distributed Arithmetic algorithm.

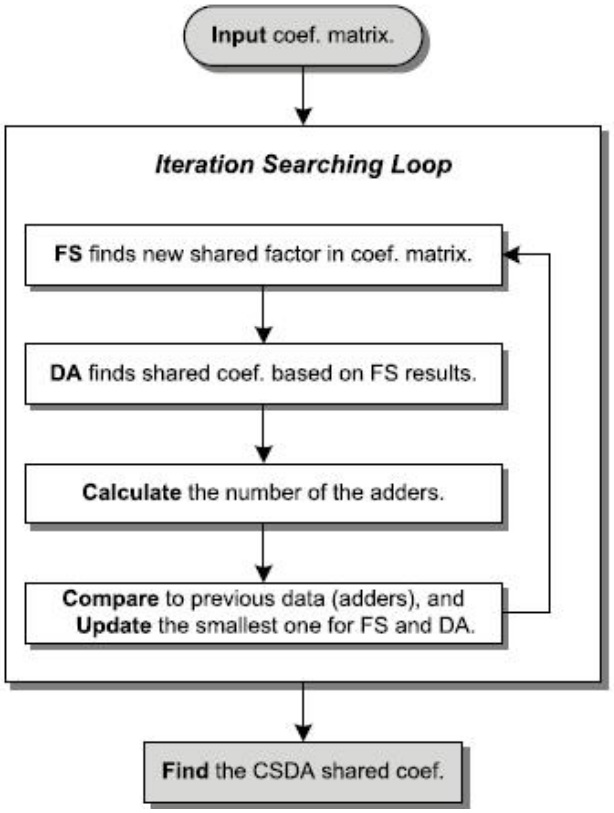


Fig.5. The Proposed CSDA searching Flow.

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.

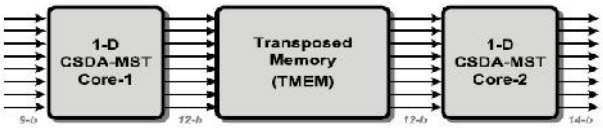
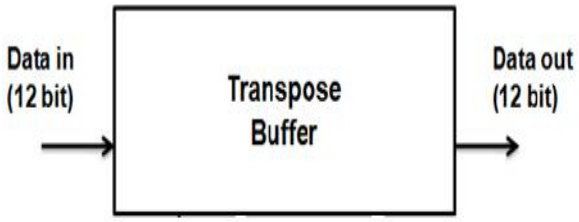


Fig.6.Proposed 2-D CSDA-MST core.

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

$$\begin{bmatrix} Z_0 \\ Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \\ Z_6 \\ Z_7 \end{bmatrix} = C \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \quad (7)$$

Where

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

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, **Z***e* and **Z***o*, as listed in (8) and(9),respectively

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

$$= C_e.a \qquad (8)$$

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

$$= C_o.b \qquad (9)$$

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 **Z***e* can be further decomposed into even and odd parts: **Z***ee* and **Z***eo*

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

$$= C_{ee}.A \qquad (11)$$

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

$$= C_{eo}.B \qquad (12)$$

Where

$$\mathbf{A} = \begin{bmatrix} a_0 + a_3 \\ a_1 + a_2 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} a_0 - a_3 \\ a_1 - a_2 \end{bmatrix}. \qquad (13)$$

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, Z_0 can be obtained by executing c4 $A0$ +$c4A1$.

$$\begin{aligned} Z0 &= c4A0 + c4A1 \\ &= (M_1\, 2^{-2} + M_2\, 2^{-5} + M_1\, 2^{-7})\, A_0 \\ &\quad + (M_1\, 2^{-2} + M_2\, 2^{-7} + M_1\, 2^{-7})\, A1 \\ &= D_{ee0}\, 2^{-2} + D_{ee1}\, 2^{-5} + D_{ee0}\, 2^{-7} \\ &= DA_{e1} 2^{-2} + DA_{e0} 2^{-5} + DA_{e1} 2^{-7} \end{aligned}$$

Where

$$DA_{e0} = D_{ee1} = (A_0 + A_1)\, M_2$$
$$DA_{e1} = D_{ee0} = (A_0 + A_1)\, M_1$$

The nonzero DA factors DA_0 and DA_{e1} 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

$$[T_0\ T1\ T2\ \ldots\ldots T_7] = [Z_0\, Z_1\, Z_2 \ldots\ldots\ldots\ldots Z_7] \qquad (14)$$

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.

CONCLUTION:

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.

K.Babu did B.Tech ECE and Persuing M.Tech VLSI at Srinivasa institute of Science and Technology (SRTS), Kadapa, AP, and doing project on High-Throughput Multistandard Transform core Supporting H.264/MPEG/VC-1 using Common Sharing Distributed Arithmetic

P.Pavan Kumar Reddy M.Tech Assistant professor Department of Electronics & Communication Engineering Srinivasa Institute of Science and Technology .Kadapa.AP.

Panem Charan Arur.He did M.Tech (VLSI System Design) and B.Tech(ECE).Now working as a Assistant Professor in ECE department at Chilkur Balaji Institute of Technology (CBIT), Hyderabad, AP,India. Doing Research Work on Low Power VLSI. Published Nine Inter National Journal, Attended Three Inter National conference and Three national level conference and two national level technical seminars,two national level workshops. Nine Professional Association member ships IAENG,CSIT,IACSIT. He has a review and Editorial,Advisory committee member in Eight International Journals.Now he doing research on advanced technologies in VLSI and Embedded systems. contact:8008179453, Email:panem.charan@gmail.com.