Analysis and Implementation of Multistandard Transform Core for MPEG/H.264/VC-1 Video Codecs using Reconfigurable Hardware Structure

Prof. Mrs. P. U. Chati[1], Ragini A. Jibhakate[2]

Dept. of electronics & Telecommunication Engg, Priyadarshini College of Engg. Nagpur, India

Abstract— In this project our main objective is to propose high throughput multi-standard transform core, which supports MPEG 1/2/4 (8 × 8), H.264 (8 × 8 & 4 × 4), and Video Codecs VC-1 (8 × 8, 8 × 4, 4×8 & 4×4) transforms. Common sharing distributed arithmetic (CSDA) consists of factor sharing(FS) and distributed arithmetic(DA) sharing techniques, efficiently reducing the number of hardware(adders) resources for high hardware resource-sharing capability in order to exploit the available resources on FPGAs. This incorporates parallel processing and pipelining of the input samples. This architecture helps to increase the throughput rate of the design. CSDA algorithm reduces nonzero elements. In the proposed MST the reduction in adders is achieved up to 44.5%, as compared with the direct implementation method. This MST core has an eightfold operation frequency throughput rate with eight parallel computation paths. Measurement shows that designed MST core has throughput rate of 1.3 G-pels/s .

Keywords— Factor sharing and distributed arithmetic (FS & DA), multi-standard transform (MST), discrete cosine transform (DCT), FPGA.

I. INTRODUCTION

Different transforms are widely used in image and video compression applications. Various groups, which are The International Organization for Standardization (ISO), Microsoft Corporation and International Telecommunication Union Telecommunication Standardization Sector (ITU-T), have developed different Transform dimensions and coefficients, corresponding to several applications (Table I).

Groups	Video Standards	Dimensions
ISO	MPEG-1/2/4	8 × 8
ITU-T	H.261, H.263	8 × 8
	H.264	8 × 8, 4 × 4
Microsoft	VC-1	8 × 8, 4 × 8, 8 × 4, 4 × 4

Common sharing distributed arithmetic (CSDA): This algorithm combines factor sharing and distributed arithmetic techniques, efficiently reduces the number of adders for high hardware-sharing capability (44.5%).

Discrete Cosine Transform:

It represents the finite sequences of data points in terms of sum of cosine term functions oscillating at different frequencies. 2D DCT is often used in signal and image processing because of its strong energy compaction property.

Number of researchers have worked on various transform core designs, that includes discrete cosine transform (DCT) as well as integer transform, which uses distributed arithmetic (DA), and factor sharing (FS) , and matrix decomposition methods for reducing hardware cost. The inner product can be implemented using ROMs and accumulators at place of multipliers that reduces the area cost. Yu and Swartzlander present very efficient method to reduce size of ROMs with DCT algorithms. Delta matrix is used to share hardware resources using the FS method. They generates matrices for multistandards as linear combinations obtained from the same matrix and delta matrix, and show that by factorization the same matrix coefficients can share the same hardware resources.

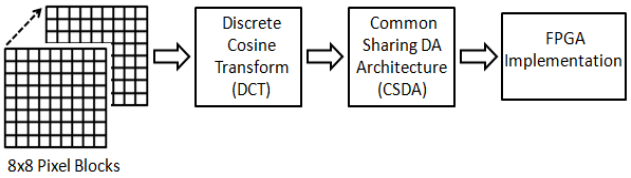
Recently, reconfigurable architecture has been proposed as a solution that achieves very good flexibility of processors in field-programmable gate array (FPGA) platform or ASIC that is application-specific integrated circuit.

In this paper we have proposed a MST core which supports MPEG-1/2/4 (8 × 8), H.264 (8 × 8 & 4 × 4), and Video Codes VC-1 (8 × 8, 8 × 4, 4 × 8 & 4 × 4) transforms. However, the proposed MST core includes DA and FS schemes which are combined as common sharing distributed arithmetic (CSDA) that reduces the hardware cost. The main aim is to reduce the nonzero elements using CSDA algorithm; which results in, few requirement of adders in the adder-tree circuit. According to the designed strategy, the selected canonic signed digital (CSD) coefficients can achieve very good sharing capability for hardware resources.

The paper is represented as follows. Section II shows the proposed plan of the work. Section III provides the mathematical derivation of the proposed CSDA algorithm.

Section IV addresses the designed 2-D CSDA-MST architecture, that includes derivation of eight-point and four point transforms, choice of coefficients, and the proposed 2-D CSDA architecture. Section V presents comparisons and discussions. Finally, Section VI offers a detailed conclusion.

II. PROPOSED METHOD



III. MATHEMATICAL DERIVATION OF THE PROPOSED CSDA ALGORITHM

The proposed CSDA combines the FS and DA methods to gain better resource sharing for inner-product operation. The FS and DA method are described as follows.

A. *Mathematical Derivation of Factor Sharing*

In Factor Sharing method the same factor is shared in different coefficients among the same input. Now, consider two different elements $S1$ and $S2$ having the same input X as an example

$$S1 = C1X,\ S2 = C2X. \tag{1}$$

Assuming that the coefficients $C1$ and $C2$ having same factor Fs can be found in the, (1) can be given as follows:

$$S1 = (Fs2^{k1} + F_{d1})X$$

$$S2 = (Fs2^{k2} + F_{d2})X \tag{2}$$

where $k1$ and $k2$ specifies the weight position of the shared factor Fs in coefficients C_1 and C_2, respectively. The remainder coefficients are $Fd1$ and $Fd2$ after extracting the shared factor Fs for coefficients C_1 and C_2, respectively

$$F_{d1} = C_1 - Fs2^{k1}$$

$$F_{d2} = C_2 - Fs2^{k2}.$$

B. *Mathematical Derivation of CSD Format Distributed Arithmetic*

For a general matrix multiplication-and accumulation the inner product can be written as follows:

$$Y = \mathbf{A}^T\mathbf{X} = \sum_{i=1}^{L} A_i X_i \tag{4}$$

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

$$Y = \begin{bmatrix} 2^0 & 2^{-1} & \cdots & 2^{-(N-1)} \end{bmatrix} \cdot \begin{bmatrix} A_{1,0} & A_{2,0} & \cdots & A_{L,0} \\ A_{1,1} & A_{2,1} & \cdots & A_{L,1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{1,(N-1)} & A_{2,(N-1)} & \cdots & A_{L,(N-1)} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_L \end{bmatrix}$$
$$= \begin{bmatrix} 2^0 & 2^{-1} & \cdots & 2^{-(N-1)} \end{bmatrix} \begin{bmatrix} Y_0 \\ Y_1 \\ \vdots \\ Y_{(N-1)} \end{bmatrix} \tag{5}$$

where $Yj = \sum^{L} A_{i,j}\ X_i$, $A_{i,j} \in \{-1, 0, 1\}$, and $j = 0, \ldots (N - 1)$.

In (5), Y_j can be computed by adding or subtracting X_i with $A_{i,j} \neq 0$. The resultant product Y can then be obtained by shifting and adding every nonzero Y_j. In this manner, the inner product computation in (4) and can be implemented with the help of shifters and adders instead of multipliers. Therefore, CSD DA-based architecture can achieve a efficient area optimization.

C. *Designed Common Sharing Distributed Arithmetic (CSDA) Algorithm*

The proposed CSDA algorithm combines the Factor Sharing (FS) and Distributed Arithmetic (DA) methods.

$$\begin{aligned} C_{11} &= [1\ -1\ 1\ 0\ 0] \\ C_{12} &= [1\ -1\ 0\ 0\ 1] \\ C_{21} &= [1\ 1\ -1\ 0\ 0] \\ C_{22} &= [0\ 1\ -1\ 0\ 0]. \end{aligned} \tag{7}$$

The coefficients matrix is expanded at the bit level then in each coefficient the same factors are shared by the FS method; after that the DA method is applied to share the same input combination for each coefficient position. The proposed CSDA algorithm example in a matrix inner product is given as follows:

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

Where $C11$ and $C22$ are the coefficients of five-bit CSD numbers

Fig. 1 shows the flow of proposed CSDA algorithm. In these four coefficients the same shared factor Fs is [1 −1], and $C11 \approx C22$ can use Fs instead of [1 −1], with the corresponding position under the FS method. To share the same position for the input, and the DA shared coefficient DA_1 which is equal to $(X_1 + X_2)\ Fs$ the DA method is subsequently applied. Finally, the matrix inner products in (6) can be performed by shifting and adding every nonzero

weight position. Fig. 2 shows the coefficient searching flow of the proposed CSDA algorithm. The FS method is adopted first to find the factors that have higher hardware resource sharing capability, In this paper the hardware resource is defined as the use of number of adders. Next, the DA method is used to search the shared coefficient based on the results of the Factor Sharing method. The adder-tree circuits will be followed by the designed CSDA circuit. For estimating the number of adders in the CSDA loop the CSDA shared coefficient is used.

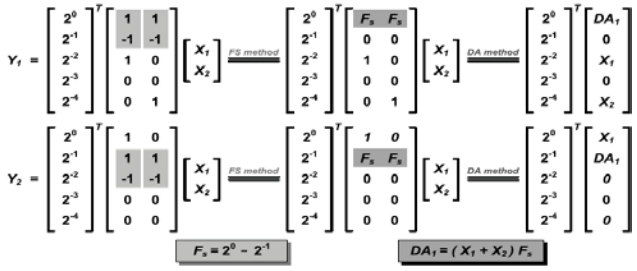


Fig.1. Example of a CSDA algorithm

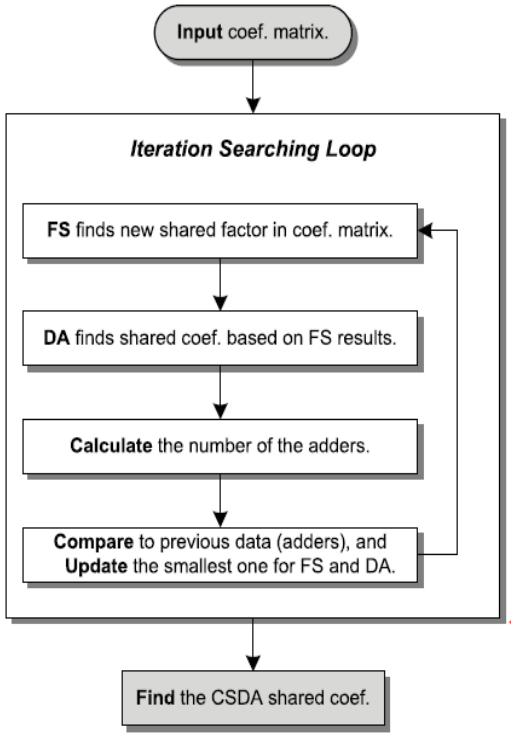


Fig.2. Illustration of the proposed CSDA searching flow.

Therefore, the iteration searching loop requires a large number of iteration loops to determine the smallest hardware resource with the help of these steps, and the CSDA shared coefficient can be generated. An example of the performance in FS, DA, and CSDA methods is shown as below. Although the joint DA and sub-expression sharing method, which adopts DA method first for eliminating multipliers and then to reduce the number of adders by the sub-expression sharing method for sharing the same adder operations, is presented in, the proposed CSDA method adopts the FS method first to share the same adder operations and then adopts the DA method to reduce the nonzero term, which can reduce the operation of the adder tree. Therefore, the proposed CSDA improves the sharing capacity and is further adopted to implement multiple block sizes and coefficients.

IV. PROPOSED 2-D CSDA-MST CORE DESIGN

This section introduces the 2-D CSDA-MST core implementation. Because of the eight-point coefficient structures in MPEG-1/2/4, H.264, and Video Codecs VC-1 standards, the eight-point transformation is compatible with same mathematic methods. From the symmetry property, the 1-D eight point transform is divided into even and odd two four-point transforms, **Z***e* and **Z***o*.

Based on the proposed CSDA algorithm, the coefficients for MPEG-1/2/4, H.264, and Video Codecs VC-1 transforms are selected to achieve high hardware sharing capability for arithmetic resources to adopt the flow of searching. Software code will be helpful for the iterative searching loops by fixing a constraint with minimum nonzero elements. Here, the constraint of minimum nonzero elements is set to be five.

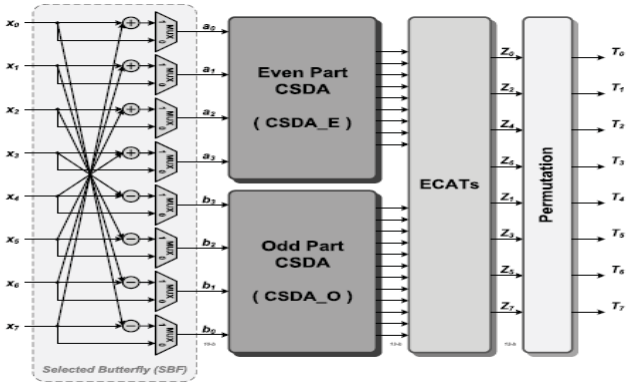


Fig.3. Architecture of the proposed 1-D CSDA-MST.

The 1-D eight-point MST core architecture is shown in Fig. 3, which consists of a selected butterfly (SBF) module, an even part CSDA (CSDA_E) and an odd part CSDA (CSDA_O), a permutation module and eight error-compensated error trees (ECATs).

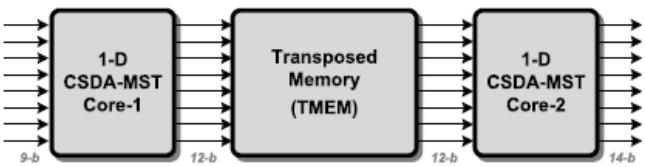


Fig.4. Architecture of proposed 2-D CSDA-MST core.

The designed 2-D CSDA-MST core consists of two 1-D CSDA-MST core with Core-1 and Core-2 having a transposed memory (TMEM), as shown in Fig. 4. Core-1

and Core-2 have different word length for arithmetic, register, and MUX. The TMEM is designed with the help of sixty-four 12-bit registers, and the output data from Core-1 is transposed and fed into Core-2. Both cores have four pipeline stages: two in the even and odd part of the CSDA circuit, and two in ECATs. Therefore, the proposed 2-D CSDA-MST core architecture has a latency of 158 clock cycles (= 64 + 30 + 64), and the TMEM executes transposed operation after 12 clock cycles (= 64 + 30) when 64 pixels are input.

V. DISCUSSION AND COMPARISONS

This section includes a discussion of the hardware resources and accuracy of the system for the proposed 2-D CSDA-MST core and also provides a comparison with previous works.

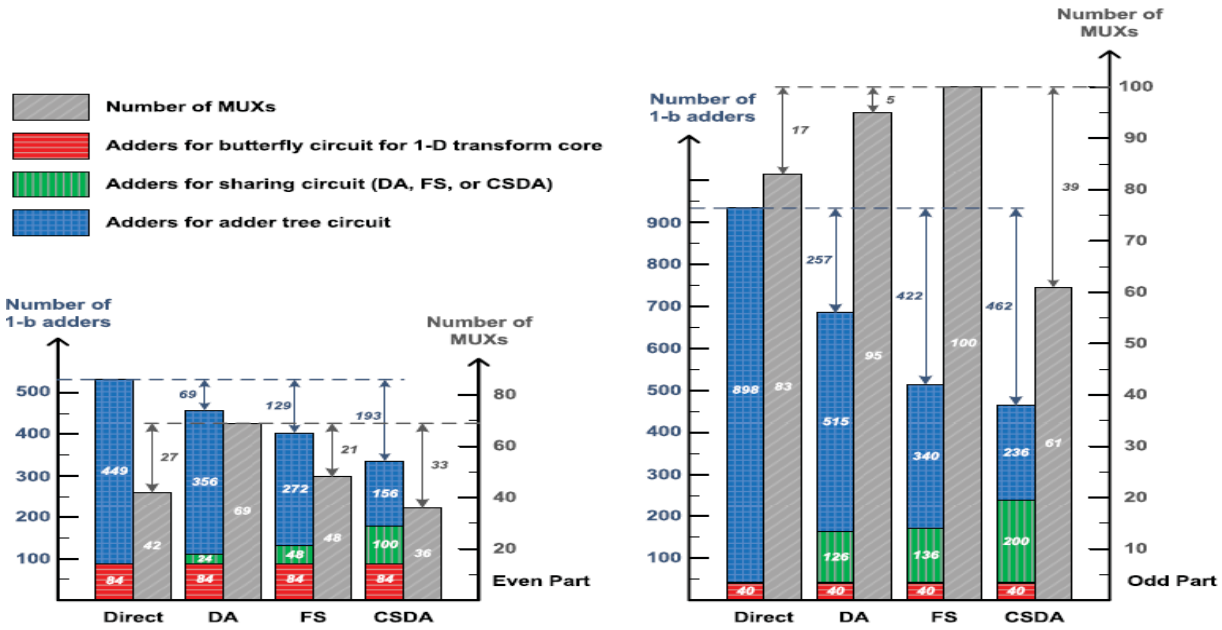


Fig. 5. Number of 1-bit adders and MUXs for the CSDA method as applied to 1-D MST core.

A. *Hardware Resources Evaluation for Proposed CSDA Method*

Fig. 5 gives a summary of the number of adders and MUXs used among direct implementation, DA, FS, and the proposed CSDA methods when applied to the 1-D MST core. The usage of adders is normalized to the 1-bit adder for a fair comparison. The direct implementation method simply replaces multipliers with adders. The 1-D MST core includes 1471 one-bit adders for the direct implementation method, which consists of 533 one-bit adders for the even part and 938 one-bit adders for the odd part. The DA or FS method is adopted in the 1-D MST core to reduce

the number of adders; 326(= 69 + 257) and 551(= 129 +422) 1-bit adders are reduced for these methods, improving the adder cost when compared with direct implementation. Fortunately, the proposed CSDA method can achieve a saving of 655(= 193 + 462) adders when compared with direct implementation. It can further use fast adder topologies to improve the performance, such as carry-look-ahead, parallel prefix, Wallace tree, or other fast adders. Considering another important component in hardware resource estimation, the DA method consumes the largest number of MUXs (= 69+95 =164). However, the CSDA method consumes only 97 MUXs to support multiple standards. The proposed CSDA method has excellent capability in resources sharing. Therefore, the least number of adders and MUXs are used by the proposed CSDA method, and a low-cost design can be achieved in the CSDA-MST.

VI. CONCLUSION

The CSDA-MST core provides high performance, with a high throughput rate and low-cost design, supporting MPEG-1/2/4, H.264, and Video Codecs VC-1 MSTs. By using the designed CSDA method, the number of adders, multipliers and MUXs in the MST core can be saved efficiently. As visual media technology has developed rapidly, this approach will help to increase high-resolution specifications and future needs as well.

REFERENCES

[1] High-Throughput Multi-standard Transform Core Supporting MPEG/H.264/VC-1 Using Common Sharing Distributed Arithmetic Yuan-Ho Chen, *Member*, *IEEE*,, Jyun-Neng Chen, Tsin-Yuan Chang, *Member*, *IEEE*, and Chih-Wen Lu, *Member*, *IEEE 2014*

[2] Power And Area Optimal Implementation Of 2d-csda For

Multi Standard Core Thammisetty Arun Kumar and N Suresh Babu. *Corresponding Author: Thammisetty Arun Kumar. Int. J. Elec & Electr. Eng & Telecoms. 2014

[3] Susrutha Babu Sukhavasi, Suparshya Babu Sukhavasi, O.Ranga Rao., G.Roopa Krishna Chandra, Sr Sastry K "Design of High Throughput DCT Core Design by Efficient Computation Technique " International Journal of Modern Engineering Research (IJMER) Vol.2, Issue.4, July-Aug2012

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

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

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

AUTHOR'S PROFILE

Ragini A Jibhakate has received hre Bachelor of Engineering degree in Electrical & Telecommunication Engineering from Umrer College of Engineering, Umrer in the year 2014. At present she is pursuing M.Tech. with the specialization of Electronics & Communication Engineering in Priyadarshini College of Engineering Nagpur. Her area of interest DIP, Signal and processing.

Prof. Mrs. P. U. Chati has received her Bachelor of Engineering degree in Electronics Engineering from KITS, Ramtek in the year 1999 and M. Tech (VLSI) degree from RKNEC Nagpur in 2007. At present she is working as assistant professor in Priyadarshini College of Engineering, Nagpur. Her area of interest are Signal processing and electromechanical analog circuits.