



## Common sharing distributed arithmetic method with eight parallel computation paths used effective multi standard transform core supporting the standards

### MPEG,H.264,VC-1

C.M.Vishnu Raj\* and S.Yuvaraj

SRM University, Kattankulathur, Kancheepuram-603203, Tamilnadu, India.

#### ARTICLE INFO

##### Article history:

Received: 21 June 2014;

Received in revised form:  
25 July 2014;

Accepted: 6 August 2014;

##### Keywords

Common Sharing Distributed Arithmetic-CSDA, Distributed Arithmetic-DA, Factor Sharing-FS.

#### ABSTRACT

This paper proposes a the common sharing distributed arithmetic method with less number of flip flops to reduce the power consumption by the circuit. To achieve the lesser number of Flip flops usage the registers used in the circuit to create latency will be replaced by buffers. By using the buffers in the circuit itself can create some latency efficiently than using the more number of flip flops in the circuit.

© 2014 Elixir All rights reserved

#### Introduction

This paper proposes a MST core that supports MPEG-1/2/4,H.264 and VC-1 transforms. The proposed MST core employs DA[1]-[5] and FS schemes as common sharing distributed arithmetic (CSDA) to reduce hardware cost. And the main advantage of the proposed method won't reduce the efficiency. But increases the throughput. The main strategy aims to reduce the nonzero elements using CSDA algorithm[1].thus, few adders are needed in the adder-tree circuit.

#### 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[4]. The foundations of the FS and DA method are described in the following.

#### Mathematical Derivation of Factor Sharing

The FS method shares the same factor in different coefficients among the same input. Consider two different elements  $S1$  and  $S2$  with the same input  $X$  as an example[3][5]

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

Assuming that the same factor  $F_s$  can be found in the coefficients  $C1$  and  $C2$ , (1) can be rewritten as follows

$$S1 = (Fs2k1 + Fd1)X \quad (2)$$

$$S2 = (Fs2k2 + Fd2)X \quad (3)$$

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

$$Fd1 = C1 - Fs2k1 \quad (4)$$

$$Fd2 = C2 - Fs2k2 \quad (5)$$

#### Existing System

The Existing CSDA method is the combination of two methods named Factor Sharing and Distributed Arithmetic Methods. The Factor sharing is nothing but sharing the same factors in a coefficient and reduces the total number of repeated process to consume timing. And the Distributed arithmetic is also same like factor sharing but in the distributed arithmetic method the same coefficients will be shared and reduces the

repeated operation in coefficient values. It also reduces the non-zero value check up and reduces the more time consumption. The image pixel value was taken from the basic image  $8 \times 8$  matrix transform.[3].And the values were selected by selected butterfly block and divide into even and odd values and fetch to the even and odd part to perform the operation. After that Even and odd part operation the outputs are given to the ECAT block to perform truncation operation and gives the 8-bit output. And by the use of Permutation block the output from ECAT will be reordered and gives the sequential output.

$$\begin{pmatrix} Z0 \\ Z1 \\ Z2 \\ Z3 \\ Z4 \\ Z5 \\ Z6 \\ Z7 \end{pmatrix} = C \begin{pmatrix} x0 \\ x1 \\ x2 \\ x3 \\ x4 \\ x5 \\ x6 \\ x7 \end{pmatrix}$$

Because the eight-point coefficient structures in MPEG-1/2/4, H.264, 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 (8) can be divided into even and odd two four-point transforms,  $Z_e$  and  $Z_o$ , as listed respectively

$$Z_e = \begin{pmatrix} Z0 \\ Z2 \\ Z4 \\ Z6 \end{pmatrix} \begin{pmatrix} C4 & C4 & C4 & C4 \\ C2 & C6 & -C6 & -C2 \\ C4 & -C4 & -C4 & C4 \\ C6 & -C2 & C2 & -C6 \end{pmatrix} \begin{pmatrix} a0 \\ a1 \\ a2 \\ a3 \end{pmatrix}$$

$$= C_e.a$$

$$Z_o = \begin{pmatrix} Z1 \\ Z3 \\ Z5 \\ Z7 \end{pmatrix} \begin{pmatrix} C1 & C3 & C5 & C7 \\ C3 & -C7 & -C1 & -C5 \\ C5 & -C1 & C7 & C3 \\ C7 & -C3 & C3 & -C1 \end{pmatrix} \begin{pmatrix} b0 \\ b1 \\ b2 \\ b3 \end{pmatrix}$$

= Co.b

Where

$$a = \begin{pmatrix} X0+X7 \\ X1+X6 \\ X2+X5 \\ X3+X4 \end{pmatrix}, \quad b = \begin{pmatrix} x0-x7 \\ x1-x6 \\ x2-x5 \\ x3-x4 \end{pmatrix}$$

The even part of the operation in (10) is the same as that of the four-point H.264 and VC-1 transformations. Moreover, the even part Ze can be further decomposed into even and odd parts: Zee and Zeo

$$Zee = \begin{pmatrix} Z0 \\ Z4 \end{pmatrix} = \begin{pmatrix} C4 & C4 \\ Z4 & C4 \end{pmatrix} \begin{pmatrix} A0 \\ -C4 \end{pmatrix} \quad A1$$

$$= Cee.A$$

$$Zeo = \begin{pmatrix} Z2 & C2 \\ Z6 & C6 \end{pmatrix} = \begin{pmatrix} C6 & B0 \\ -C2 & B1 \end{pmatrix}$$

$$= Ceo.B$$

Where

$$A = \begin{pmatrix} a0 + a3 \\ a1 + a2 \end{pmatrix}, \quad B = \begin{pmatrix} a0 - a3 \\ a1 - a2 \end{pmatrix}$$

**Proposed 2-D CSDA-MST Core Design**

I introduces the proposed 2-D CSDA-MST core with the buffers instead of pipelined registers. Basically register is the combination of two or more flip flops. So, by replacing the registers by buffers the number of flip flops utilization will be reduced by approximately 40 % to 50 %. And by connecting the two 1-D CSDA core using the Transposed memory will Results the Proposed 2D CSDA Core.

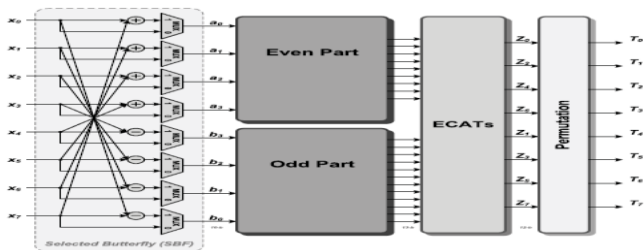
**1-D Common Sharing Distributed Arithmetic-MST:**

The 1D CSDA MST core is the combination of five main part. They are :

- a. Selected Butterfly.
- b. Even Part.
- c. Odd Part.
- d. Error Compensation Adder Tree.
- e. Permutation Block.

**Selected Butterfly:**

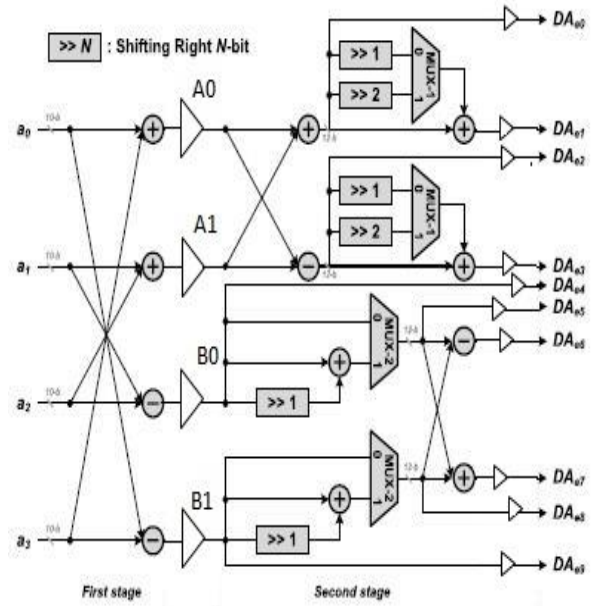
The Selected butterfly module is the combination of 8-Multiplexers, 4- Summers, and 4-Subtractors. It executes the eight-point transform and bypasses the input data for two four-point transforms. The outputs from the SBF will be denoted as 'a' and 'b'. These outputs will be fetch to the even part and odd part.



**Figure 1. 1-D Common Sharing Distributed Arithmetic-MST**

**Even Part:**

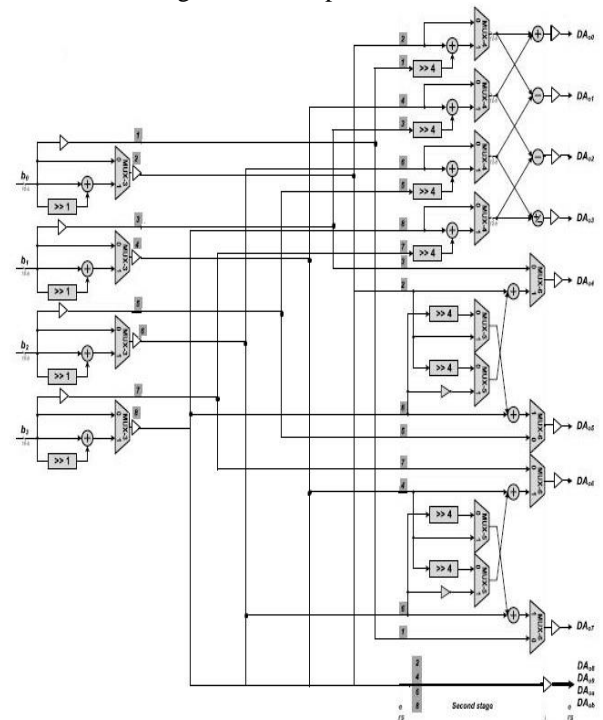
The Even Part will calculates the even values or the even part of the eight point transform. This calculation is similar to the four point transform for the image standards.



**Figure 2. Even Part Common Sharing Distributed Arithmetic Circuit**

**Odd Part**

Similar to the Even Part it will calculates the odd values (or) the odd part of the eight point transform. This calculation is similar to the four point transform for the image standards. It contains selection signals of multiplexers for different standards.



**Figure 3. Proposed Odd Part Common Sharing Distributed Arithmetic Circuit**

**Error Compensation Adder Tree**

Eight adder trees with error compensation (ECATs) [6] are followed by the Even Part and Odd Part, which add the nonzero CSDA coefficients with corresponding weight as the tree-like

architectures. It can alleviate truncation error efficiently in small area design when summing the nonzero data all together.

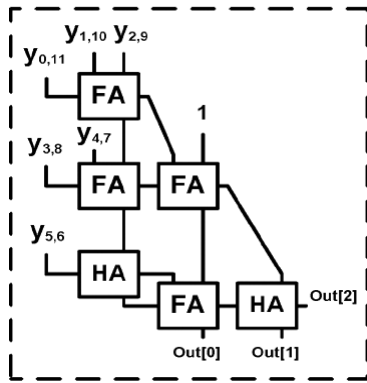


Figure 4. ECAT Architecture

**Permutation Block**

The Permutation block is used to reordering the output from the ECAT and give the output in sequential order.

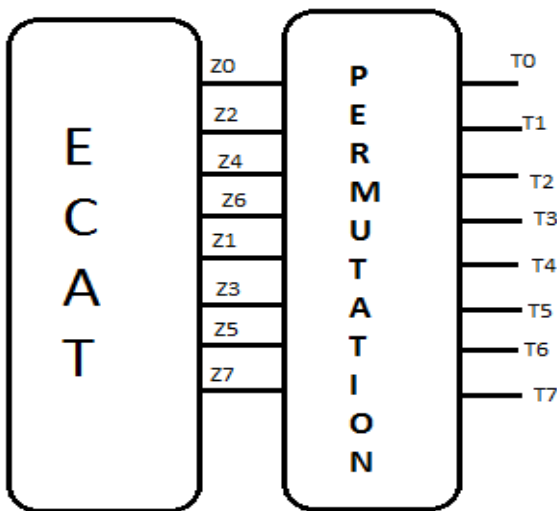


Figure 5. Permutation Block

**Transposed Memory**

This is the main block to combine two 1D CSDA blocks to get the proposed 2D CSDA MST Core. By using the Transposed memory the output from the first 1D CSDA core will be transposed and given to the second 1D CSDA core to get the error-free output from the proposed 2D CSDA core.

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[4][8], the 1st-D and 2nd-D transforms are able to be computed simultaneously.

The transposition memory is an 8x8 register array with the data width of 16 bits and is shown in Fig.

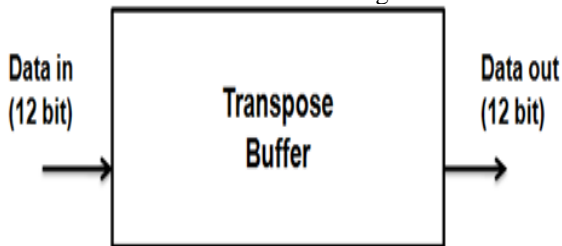


Figure 6. Transpose Buffer

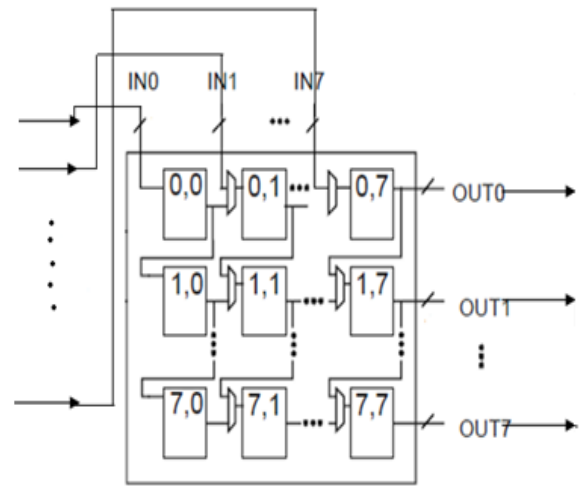


Figure 7. Circuit Diagram Of Transpose Buffer  
**2-D common sharing distributed arithmetic -MST core**

The 2-D Common Sharing Distributed Arithmetic- MST Core is the proposed method to perform the image application transformations efficiently. By Using the two FS and DA methods in the CSDA the method can support all the main image standards MPEG versions, VC-1, and H.264 versions. Comparing to other methods the proposed CSDA will support VC-1 standard also. Because of its High resolution the previous methods are not supporting the VC-1 standard. And the 2D CSDA can support for the resolution values 4928x2048.

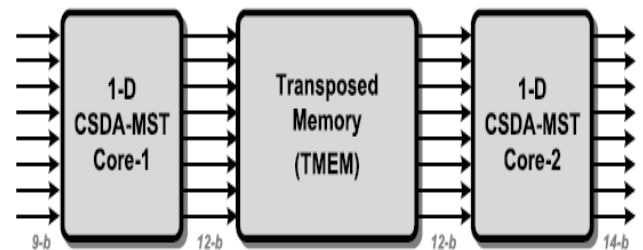
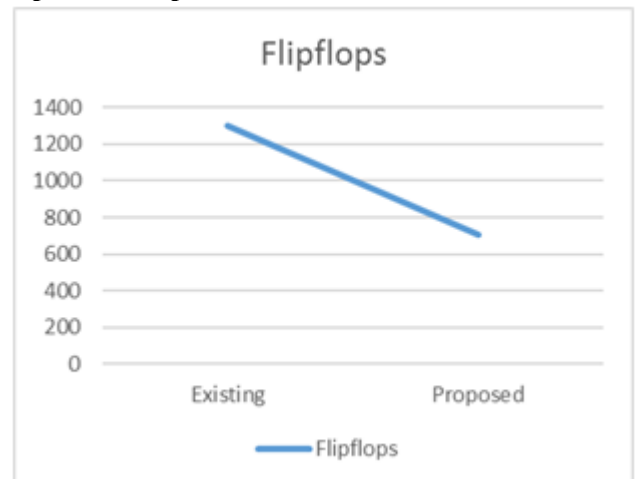


Figure 8. 2-D CSDA Core

This much of resolution range is for Ultra High resolution image operations. So the 2D CSDA will perform a wonderful role in the image applications, Digital cinemas etc.. By using the ROM based DA method the proposed 2D CSDA can reduce the circuit size by 50-80% on average.

**Device Utilization Summary**

**Comparison Graphs**



**Device Utilization Summary  
Existing System**

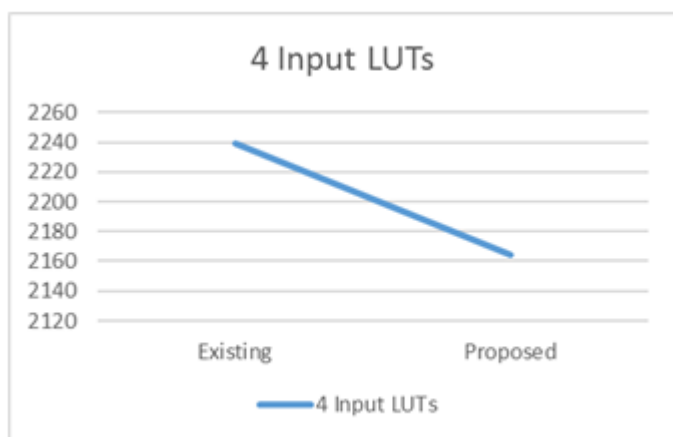
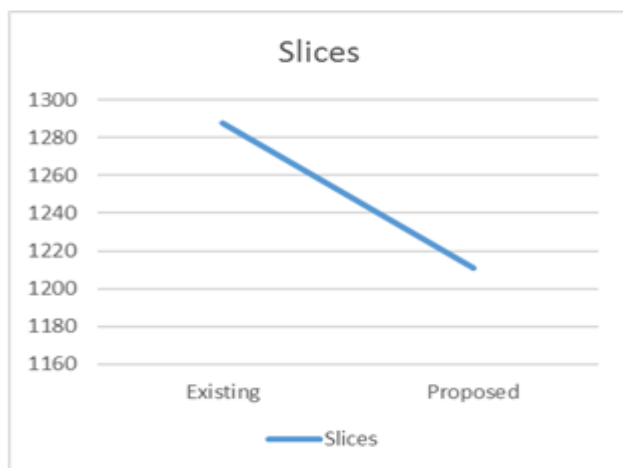
Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	1211	3584	33%
Number of Slice Flip Flops	704	7168	9%
Number of 4 input LUTs	2164	7168	30%
Number of bonded IOBs	188	141	133%
Number of GCLKs	1	8	12%

**Proposed System**

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	1288	3584	35%
Number of Slice Flip Flops	1259	7168	17%
Number of 4 input LUTs	2239	7168	31%
Number of bonded IOBs	188	141	133%
Number of GCLKs	1	8	12%

**Table 1. Comparison Table**

Components	Flip Flops	Slices	4 Input LUTs
Existing Circuit	1299	1288	2239
Proposed Circuit	704	1211	2164



### Applications

1. Video and image applications
2. Digital cinema at ultrahigh resolution

### Advantages

High-throughput rate  
Low cost

### Conclusion

The Proposed CSDA-MST core can achieve high performance, with a high throughput rate and low-cost VLSI design, supporting the standards the number of flip flops and slices are saved efficiently. Measured results show the proposed CSDA-MST core is efficiently reduces the total number of components in the circuit and efficiently reduces the area and power consumption by the circuit.

### References

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

[2] Chih-Peng Fan, Member, IEEE, and Guo-An Su, "Fast Algorithm and Low-Cost Hardware-Sharing Design of Multiple Integer Transforms for VC-1", *IEEE Transactions On Circuits And Systems—II: Express Briefs*, Vol. 56, No. 10, October 2009.

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

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] Honggang Qi, Qingming Huang, and Wen Gao, "A Low-Cost Very Large Scale Integration Architecture for Multistandard Inverse Transform" in *IEEE Trans On Circuits And Systems—II: Express Briefs*, Vol. 57, No. 7, July 2010.

[13] Tian-Sheuan Chang, Chin-Sheng Kung, And Chein-Wei Jen, "A Simple Processor Core Design For Dct/Idct" *IEEE Trans On Circuits And Systems For Video Technology*, Vol. 10, No. 3, April 2000.

[14] Kuan-Hung Chen, Jiun-In Guo, Jinn-Shyan Wang, Ching-Wei Yeh, And Jia-Wei Chen, "An Energy-Aware IP Core Design For The Variable-Length DCT/IDCT Targeting At MPEG4 Shape-Adaptive Transforms" *IEEE Trans On Circuits And Systems For Video Technology*, Vol. 15, No. 5, May 2005.