MULTI STANDARD TRANSFORM CORE USING CSDA

1. Rahul N. Khadke 2. Honrao S. B.

1. M.E. (Student), Department of Electronics and Telecommunication, AEC, Beed, Maharashtra, India.-

2. Assistant Professor and H.O.D., Department of Electronics and Telecommunication, AEC, Beed, Maharashtra, India.

ABSTRACT:

The proposed high throughput multi standard transform it performs high speed compression MPEG, VC-1, and H.264 transformation using VLSI arithmetic calculations. It is optimized compression technique derived from basic compression scheme which results low cost and excellent quality with efficiency. As it makes the use of common sharing distributed arithmetic (CSDA) technique that efficiently reduces number of adders for high hardware sharing capability. Thus CSDA-MST core achieves a high throughput rate supporting multi standard transformation at low cost.

Keywords: *Compression Techniques, VLSI design, CSDA, FPGA*

1.INTRODUCTION:

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 contents services.

A significant amount of research has been conducted to efficiently combine and implement the transform units for multiple codec's. On the other hand little research is focused on the implementation of multi-quantized unit. Among the multiple-transform units, a unified Inverse Discrete Cosine Transform (IDCT) architecture to support five standards (such as, AVS, H.264, VC-1, MPEG-2/4 and JPEG) is presented. The authors in offer an area efficient architecture to perform a DCT-based transform for JPEG, MPEG-4, VC-1 and H.264 using delta mapping. The design in is an IDCT and IQ circuit for H.264, MPEG-4 and VC-1. The MJPEG standard defines quantization as the division operation of the DCT coefficient coming from the transform unit by the corresponding Q value (specified by the quantization matrix). MJPEG allows specification of Q-matrices that facilitates the allocation of more bits for the representation of coefficients which are visually more significant.

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. In this brief, low cost very large scale integration (VLSI) architecture is designed for multi standard inverse Discrete Cosine transform. It is used in multi standard decoder of MPEG-2, MPEG-4 ASP, and VC-1 .Two circuit share strategies, factor share (FS) an adder share (AS) are applied to the inverse transform architecture for saving its circuit resource. Pipelined stages are used in this Multistandard inverse transform to increase the operational speed.

The possibility to employ hierarchical prediction structures for providing temporal scalability with several layers while improving the coding efficiency and increasing the effectiveness of quality and spatial scalable coding. New methods for inter-layer prediction of motion and residual improving the coding efficiency of spatial scalable and quality scalable coding. The concept of key pictures for efficiently controlling the drift for packet-based quality scalable coding with hierarchical prediction structures. Single motion compensation Loop decoding for spatial and quality scalable coding providing a decoder complexity close to that of single-layer coding.

2. MODULE DESCRIPTION:

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.

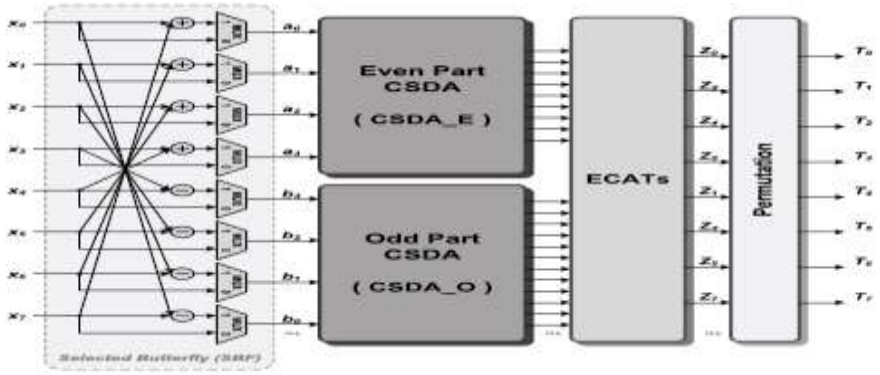

Fig-1: 1-D common sharing distributed arithmetic-MST

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.

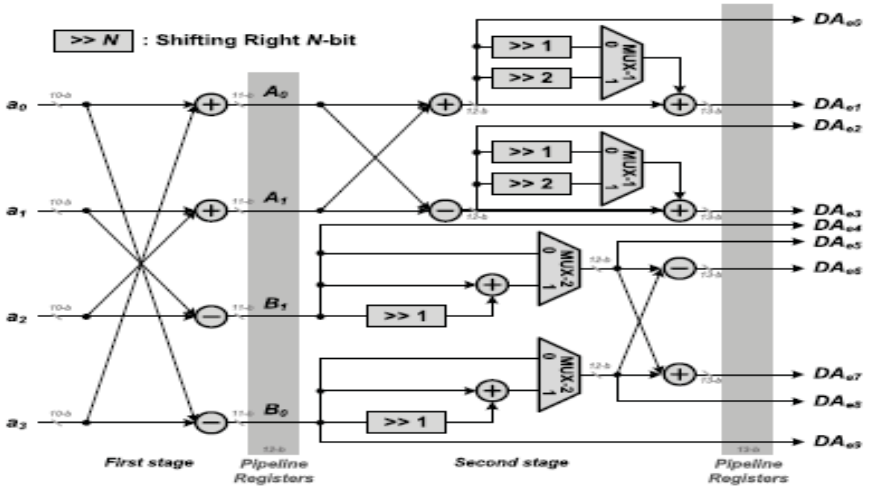

Fig-2: Even part common sharing distributed arithmetic circuit

Odd part common sharing distributed arithmetic 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.

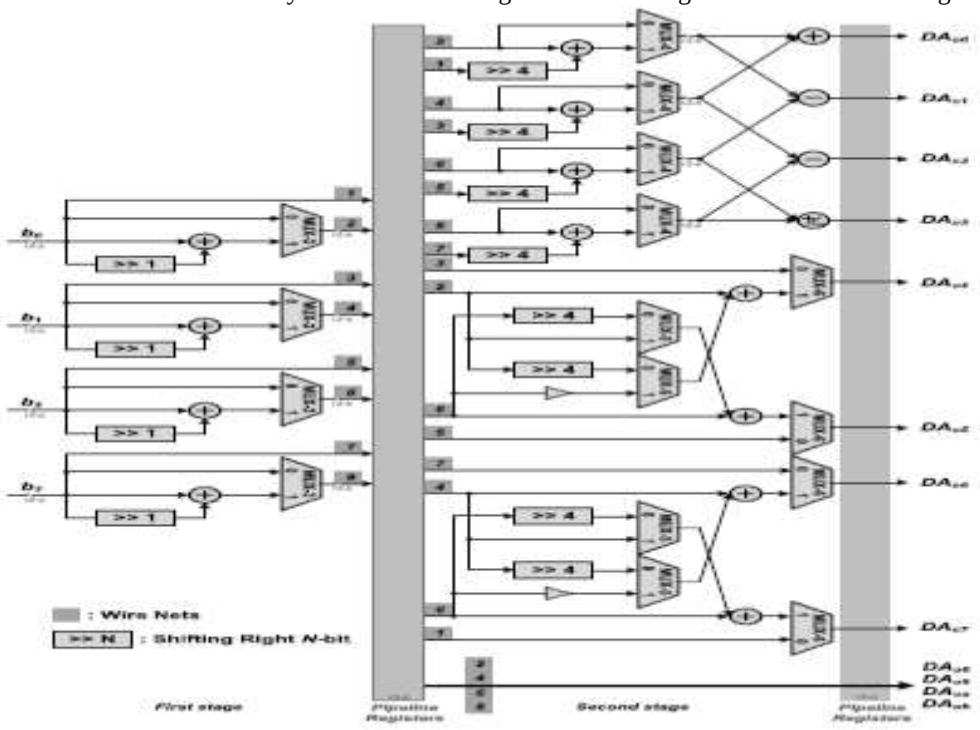

Fig-3: Odd part common sharing distributed arithmetic circuit

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.

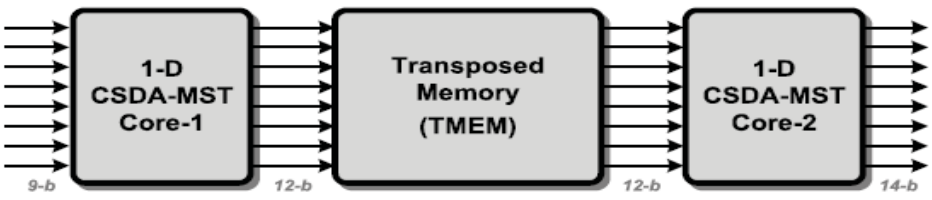

Fig-4: Block diagram of 2-D CSDA

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.

CSDA:

Distributed Arithmetic (DA) has been wildly adopted for its computational efficiency in many digital signal processing applications such as DCT (Discrete Cosine Transform), DFT (Discrete Fourier Transform), FIR (Finite Impulse Response), and DHT (Discrete Hartley Transform). These applications involve the computation of inner products between two vectors, one of which is a constant. The general method to generate the products is to use a MAC (multiply and accumulate) unit, which is fast but has a high cost in the case of long-length inner-products. In contrast, DA provides an efficient solution to realize the inner products by using memory look-up and accumulation

operations. The idea behind the conventional DA, called ROM-based, is to replace multiplication operations by pre-computing all possible values and storing these in a ROM. According to the ROM based DA can reduce the circuit size by 50-80% on average. Custom reconfigurable technology emerged to satisfy the simultaneous demand for flexibility and efficiency. Custom/domain-specific reconfigurable arrays can be programmed to adapt for different applications, so the efficiency of the hardware and flexibility of the whole system is improved. Earlier works, such as, show good performance in area, power consumption and speed. Since a domain-specific reconfigurable architecture targets few application fields, it achieves better performance than a general purpose FPGA device.

PROPOSED 2-D CSDA-MST CORE DESIGN

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

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

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 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 in (9) and (10), respectively

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

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

Where

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

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

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

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

Where

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

VLSI:

VLSI stands for "Very Large Scale Integrated Circuits". It's a classification of ICs. An IC of common VLSI includes about millions active devices. Typical functions of VLSI include Memories, computers, and signal processors, etc. A semiconductor process technology is a method by which working circuits can be manufactured from designed specifications. There are many such technologies, each of which creates a different environment or style of design. In integrated circuit design, the specification consists of polygons of conducting and semiconducting material that will be layered on top of each other to produce a working chip.

When a chip is custom-designed for a specific use, it is called an application-specific integrated circuit (ASIC). Printed-circuit (PC) design also results in precise positions of conducting materials, as they will appear on a circuit board; in addition, PC design aggregates the bulk of the electronic activity into standard IC packages, the position and interconnection of which are essential to the final circuit. Printed circuitry may be easier to debug than integrated circuitry is, but it is slower, less compact, more expensive, and unable to take advantage of specialized silicon layout structures that make VLSI systems so attractive.

FPGA:

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by the customer or designer after manufacturing—hence "field-programmable". The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC) (circuit diagrams were previously used to specify the configuration, as they were for ASICs, but this is increasingly rare). FPGAs can be used to implement any logical function that an ASIC could perform. The ability to update the functionality after shipping, partial re-configuration of the portion of the

Design and the low non-recurring engineering costs relative to an ASIC design (notwithstanding the generally higher unit cost), offer advantages for many applications.

FPGAs contain programmable logic components called "logic blocks", and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together"—somewhat like a one-chip

Programmable breadboard. Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.

Simulation Analysis:

Verilog HDL is a Hardware Description Language (HDL). A Hardware Description Language is a language used to describe a digital system, for example, a computer or a component of a computer. One may describe a digital system at several levels. For example, an HDL might describe the layout of the wires, resistors and transistors on an Integrated Circuit (IC) chip, i. e., and the switch level. Or, it might describe the logical gates and flip flops in a digital system, i. e., the gate level. An even higher level describes the registers and the transfers of vectors of information between registers. This is called the Register Transfer Level (RTL). Verilog supports all of these levels. However, this handout focuses on only the portions of Verilog which support the RTL level.

Selected Butterfly:

Fig-5- output waveform for selected butterfly

MUX:

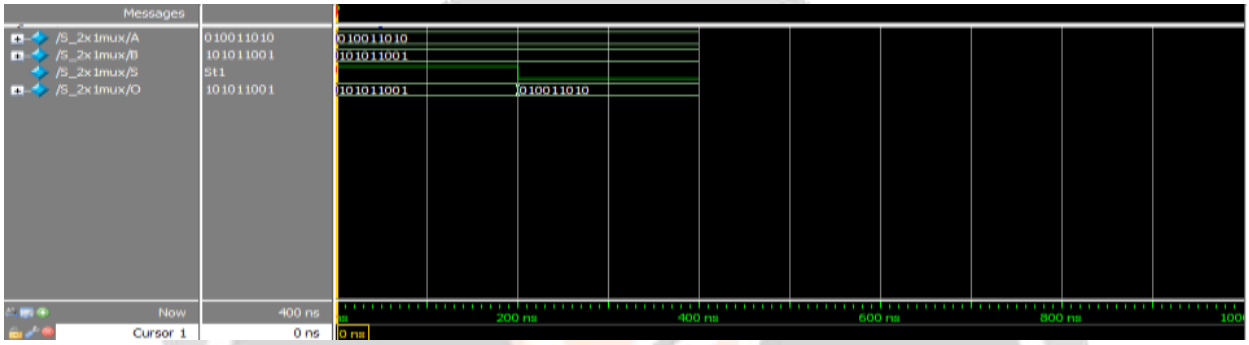

Fig-6- mux output waveform

Even Part CSDA:

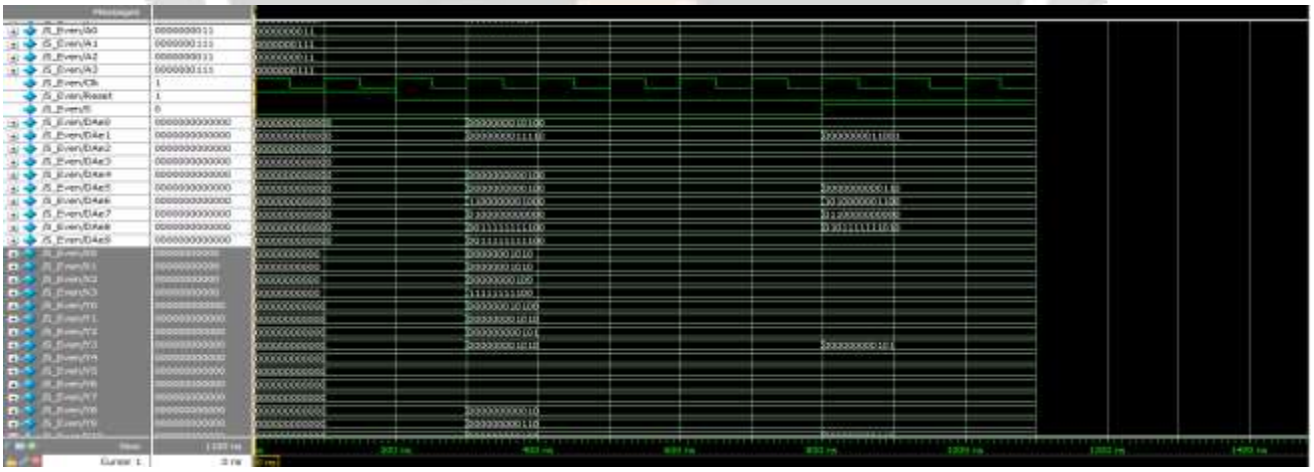

Fig7-Outputwaveform for Evenpart common Sharing distributed Arithmetic

1ST STAGE:

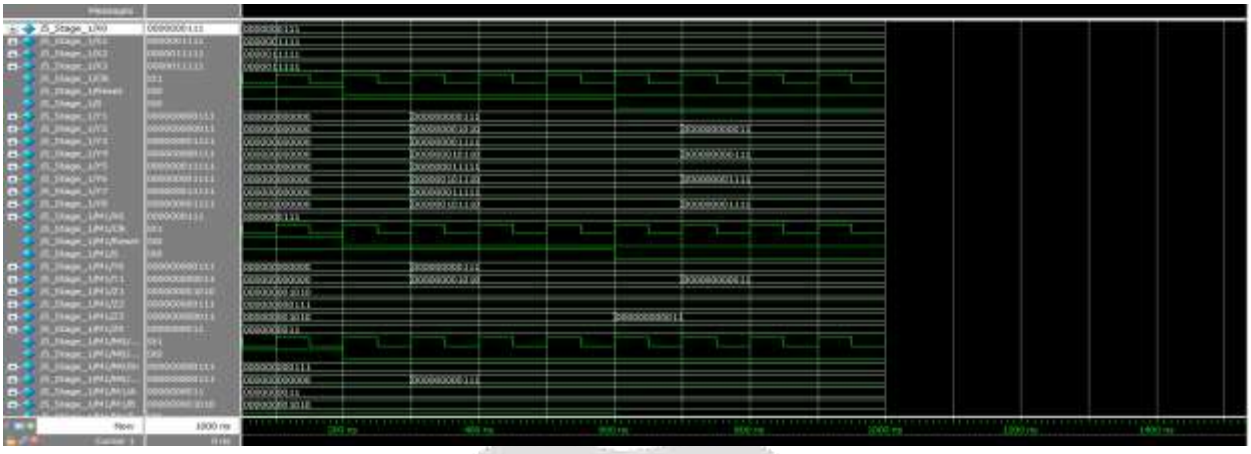

Fig 8-First stage sub module output waveform

2ND STAGE SUB0:

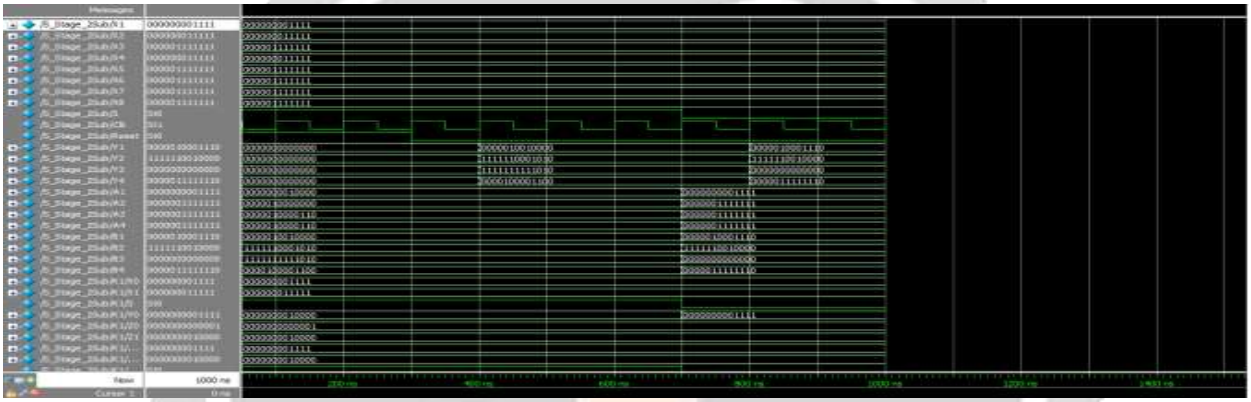

Fig 9-Second stage sub module0 output waveform

2NDSTAGE SUB2:

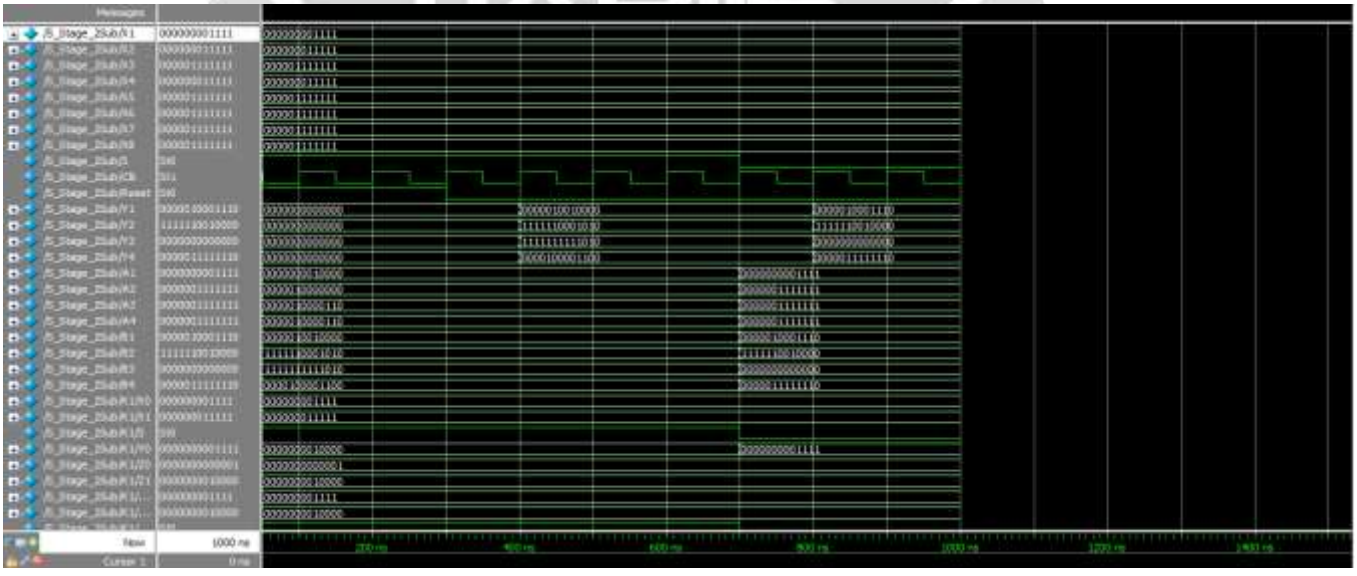

Fig 10-Second stage Sub2 module output wave form

Top Module Odd Part CSDA:

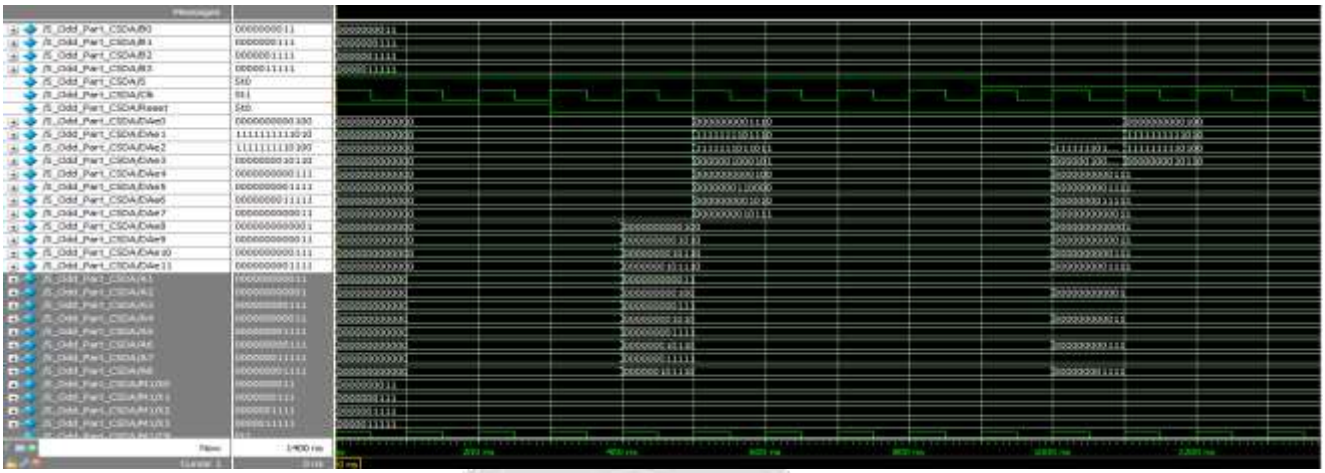

Fig 11-Top module odd part common sharing distributed arithmetic output

ECAT_1D:

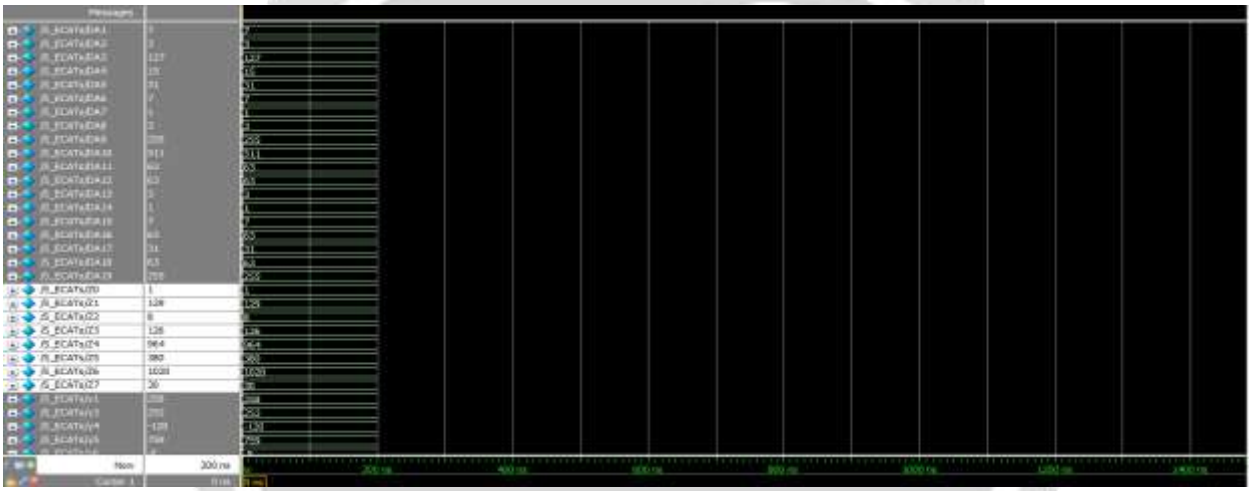

Fig 12-Output waveform of Error –compensated Adder Tree (ECAT-1D)

Permutation_1D:

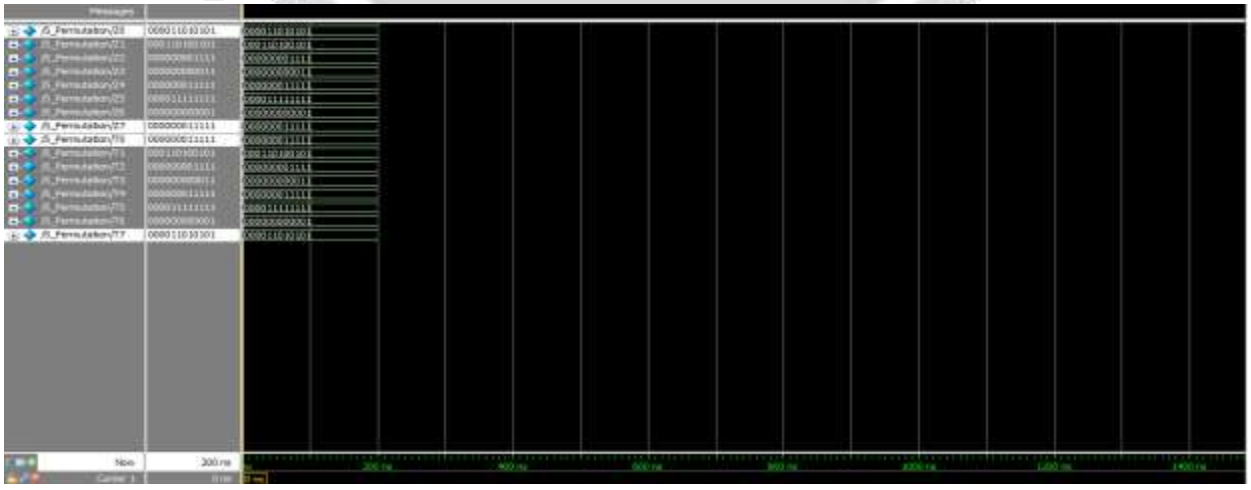

Fig 13- Permutation matrix for each of the N _N M-bit fixed point constants

1D CSDA MST 1 S=0:

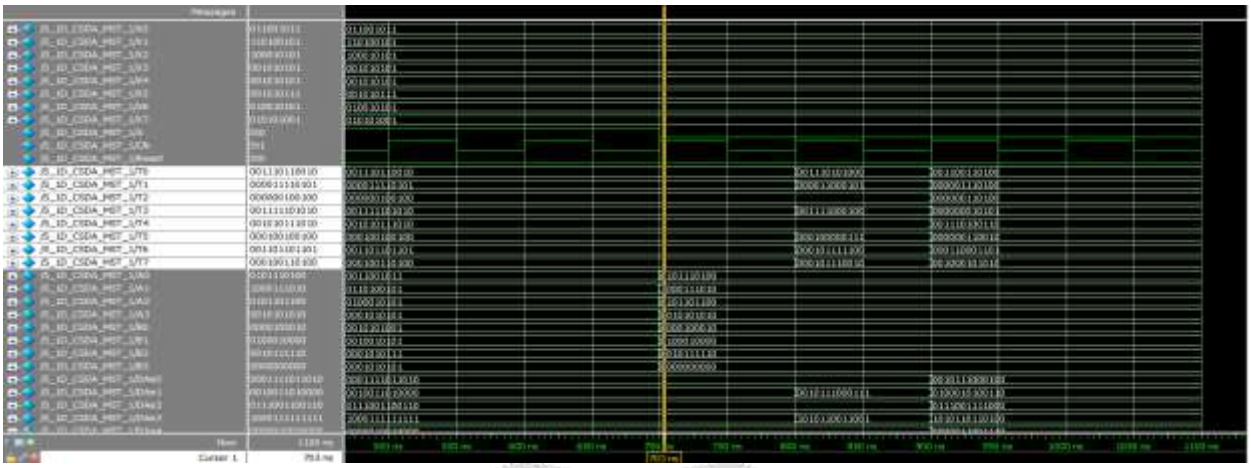

Fig 13-output waveform of one dimensional common sharing distributed arithmetic multistandard transform-1 when s=0

1D CSDA MST 1 S=1

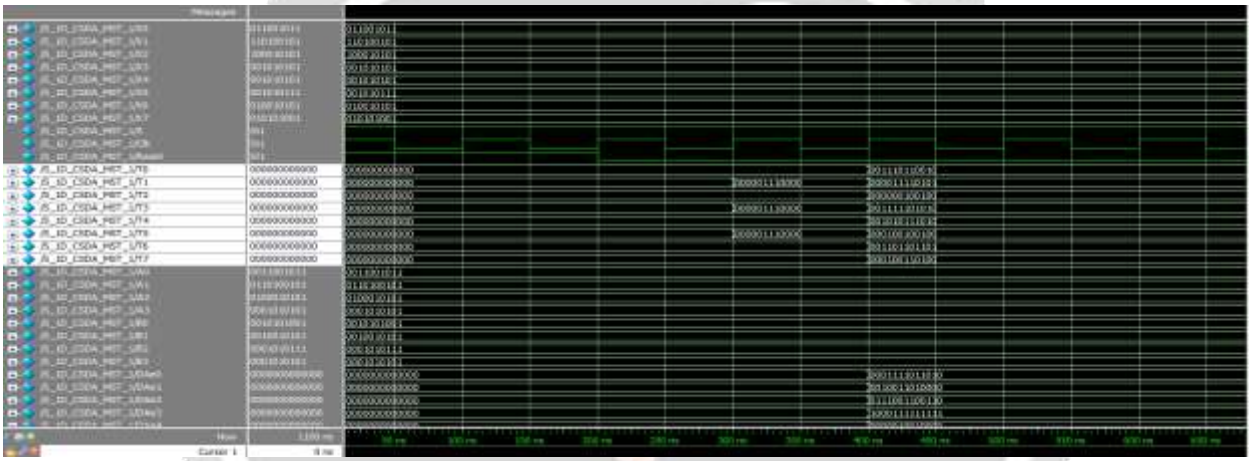

Fig 14- output waveform of one dimensional common sharing distributed arithmetic multistandard transform-1 when s=1

1D CSDA MST 2 S=0:

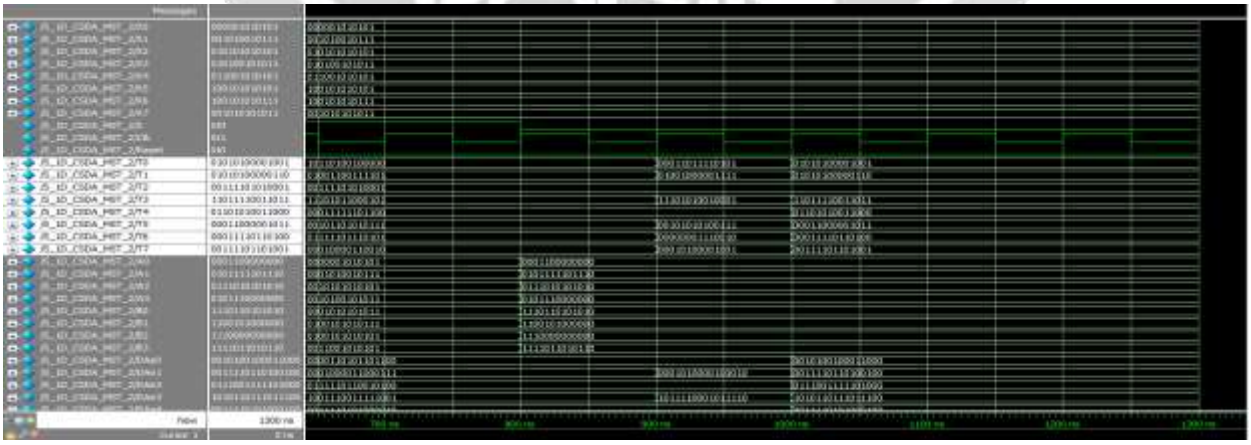

Fig 15- output waveform of one dimensional common sharing distributed arithmetic multistandard transform-2 when s=0

1D CSDA MST 2 S=1:

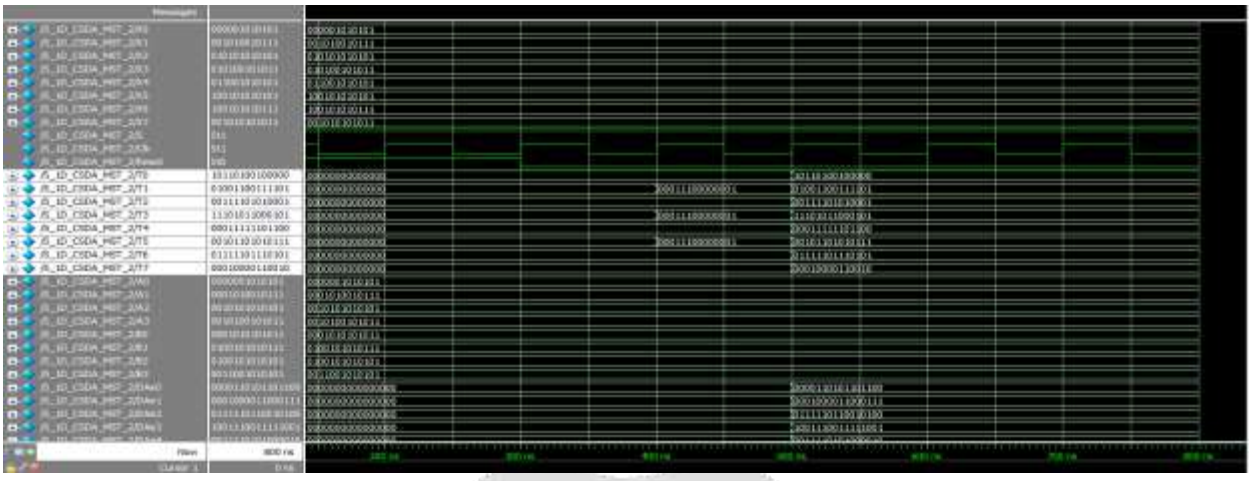

Fig 16- output waveform of one dimensional common sharing distributed arithmetic multistandard transform-2 when s=1

Transpose Memory:

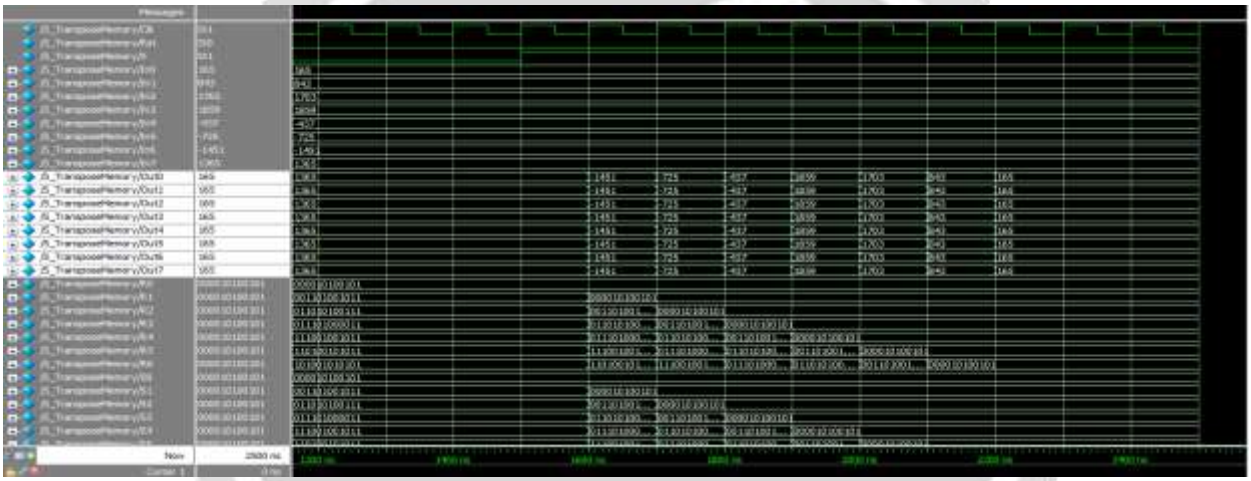

Fig 17-output waveform of Transpose memory

Top module 2D CSDA_MST CORE SelTx&Rst=0:

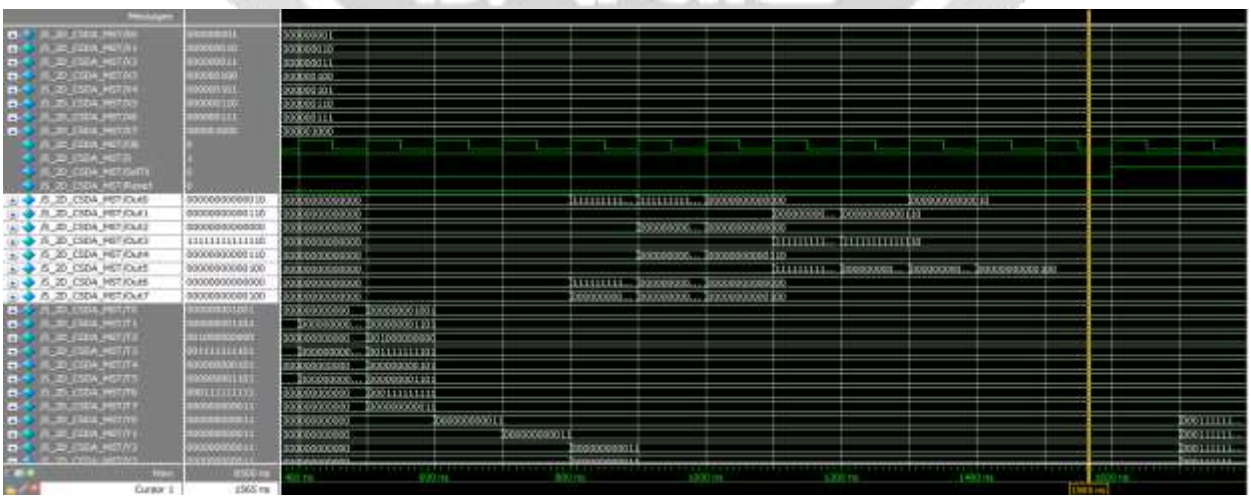

Fig 18-output waveform of 2D Common sharing distributed arithmetic for MST CORE when transmit & receive port is =0

Top module 2D CSDA_MST CORE SelTx&Rst=0 Decimal:

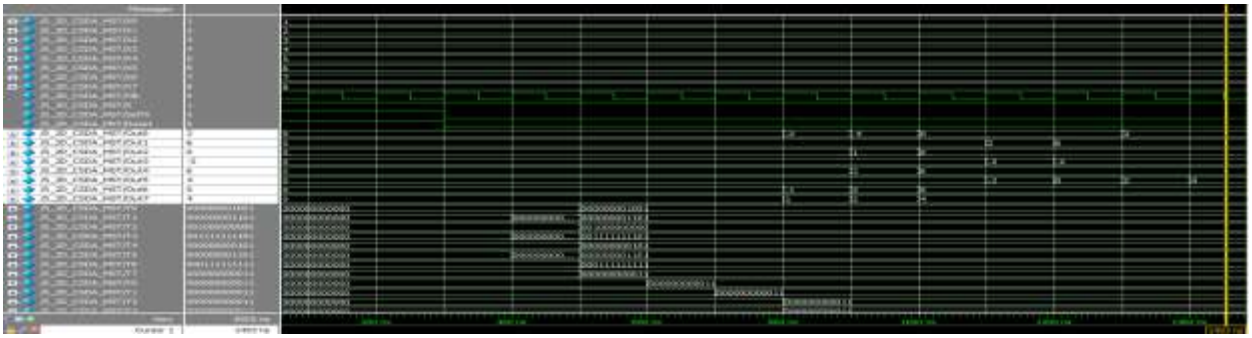

Fig 19- Output waveform of Top module 2D CSDA_MST core when SE1TX & Rst=0

Top module 2D CSDA_MST CORE SelTx=1:

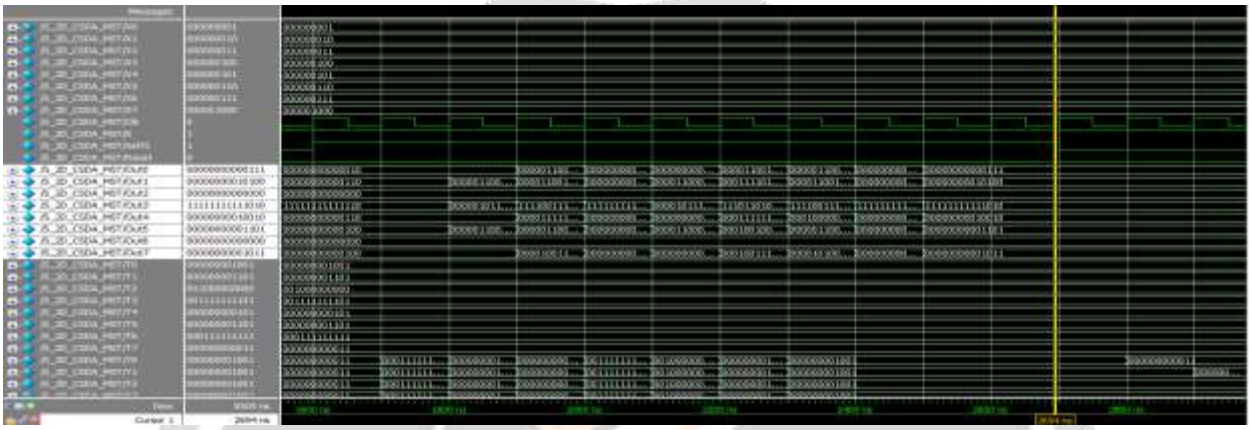

Fig 20- Output waveform of Top module 2D CSDA_MST core when SE1TX=1

Top module 2D CSDA_MST CORE SelTx=1 Decimal:

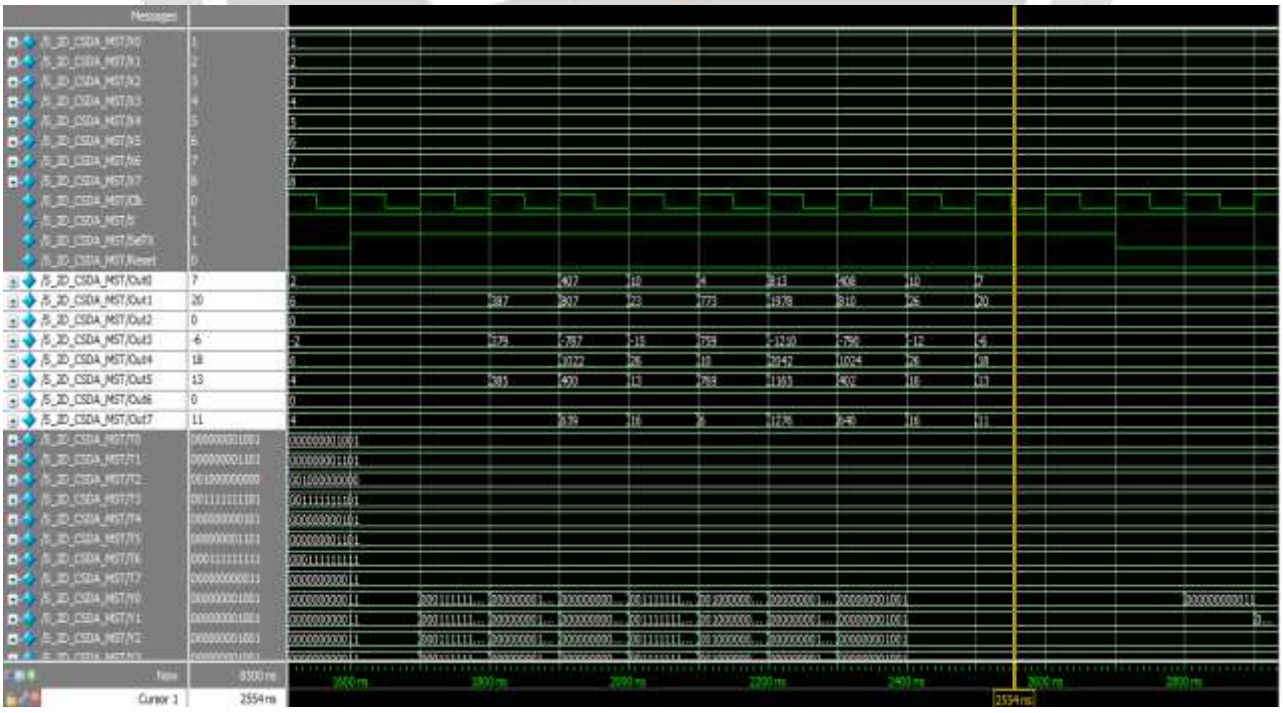

Fig 21- Output waveform of Top module 2D CSDA_MST core when SE1TX=1 Decimal

3. DEVICE UTILIZATION SUMMARY:

Proposed:

Project File:	ITVL26.ise	**Current State:**	Placed and Routed
Module Name:	S_2D_CSDA_MST	**• Errors:**	No Errors
Target Device:	xc3s400-5fg320	**• Warnings:**	513 Warnings
Product Version:	ISE 9.1i	**• Updated:**	Thu Dec 5 14:03:55 2013

ITVL26 Partition Summary

No partition information was found.

Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	1,259	7,168	17%	
Number of 4 input LUTs	2,095	7,168	29%	
Logic Distribution				
Number of occupied Slices	1,310	3,584	36%	
Number of Slices containing only related logic	1,310	1,310	100%	
Number of Slices containing unrelated logic	0	1,310	0%	
Total Number of 4 input LUTs	2,279	7,168	31%	
Number used as logic	2,095			
Number used as a route-thru	184			
Number of bonded IOBs	188	221	85%	
Number of GCLKs	1	8	12%	
Total equivalent gate count for design	28,606			
Additional JTAG gate count for IOBs	9,024			

Modification:

Project File:	modi.ise	**Current State:**	Placed and Routed
Module Name:	S_2D_CSDA_MST	**• Errors:**	No Errors
Target Device:	xc3s400-5fg320	**• Warnings:**	215 Warnings
Product Version:	ISE 9.1i	**• Updated:**	Thu Dec 5 19:03:35 2013

MODI Partition Summary

No partition information was found.

Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	704	7,168	9%	
Number of 4 input LUTs	2,048	7,168	28%	
Logic Distribution				
Number of occupied Slices	1,239	3,584	34%	
Number of Slices containing only related logic	1,239	1,239	100%	
Number of Slices containing unrelated logic	0	1,239	0%	
Total Number of 4 input LUTs	2,158	7,168	30%	
Number used as logic	2,048			
Number used as a route-thru	110			
Number of bonded IOBs	188	221	85%	
Number of GCLKs	1	8	12%	
Total equivalent gate count for design	23,799			
Additional JTAG gate count for IOBs	9,024			

RTL Schematic:

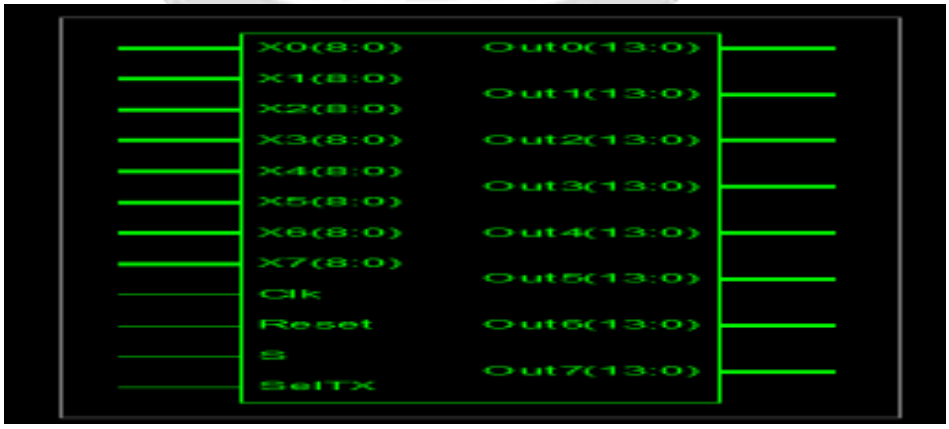

Fig: 22: S_2D_CSDA_MST

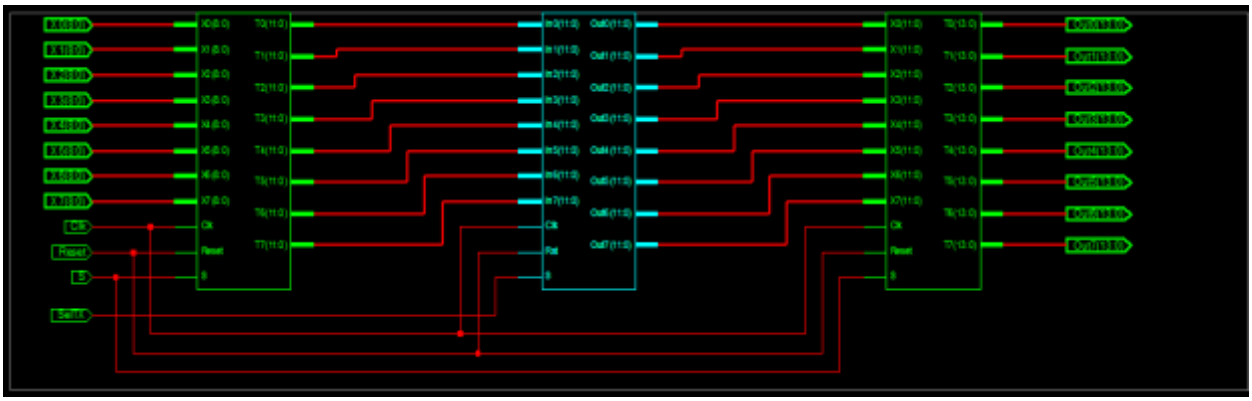

Fig: 23: CSDA_MST_2D with Transpose Memory

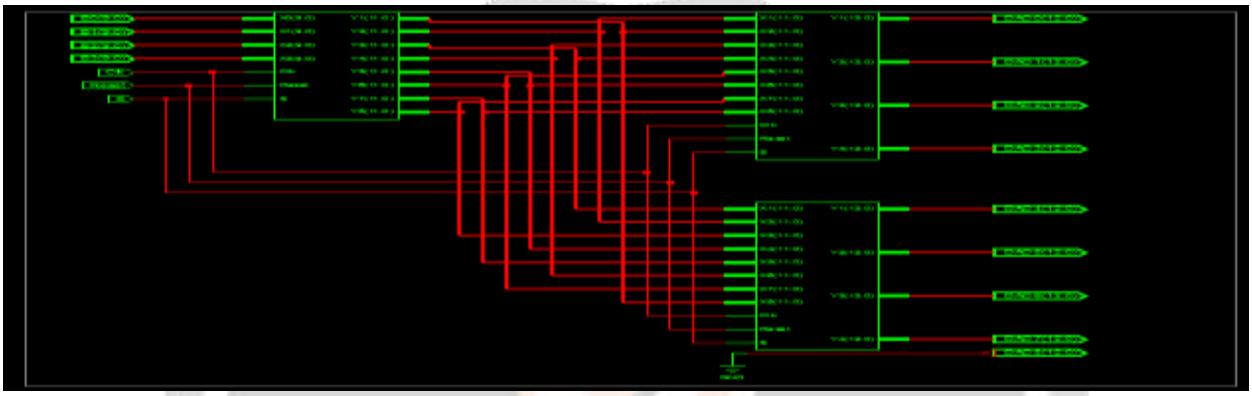

Fig 24: Internal Schematic for 2D CSDA MST

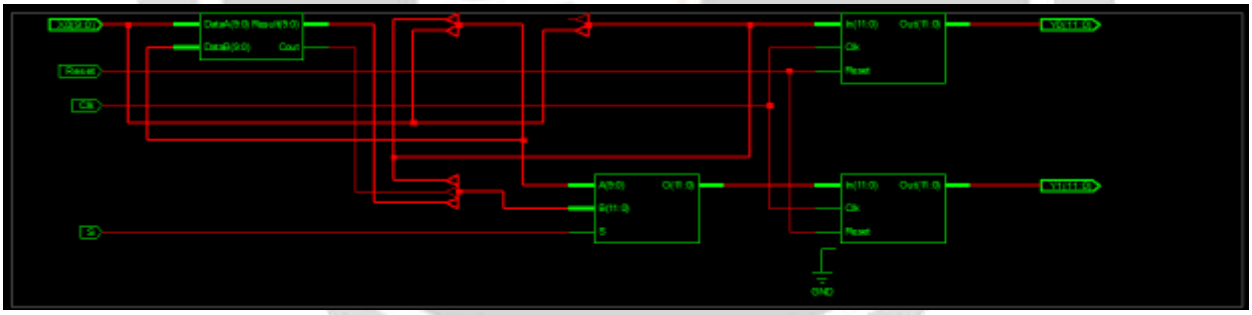

Fig 25: RTL Schematic for S_BUF

Fig 26: RTL schematic View In Xilinx for Odd part & even part CSDA

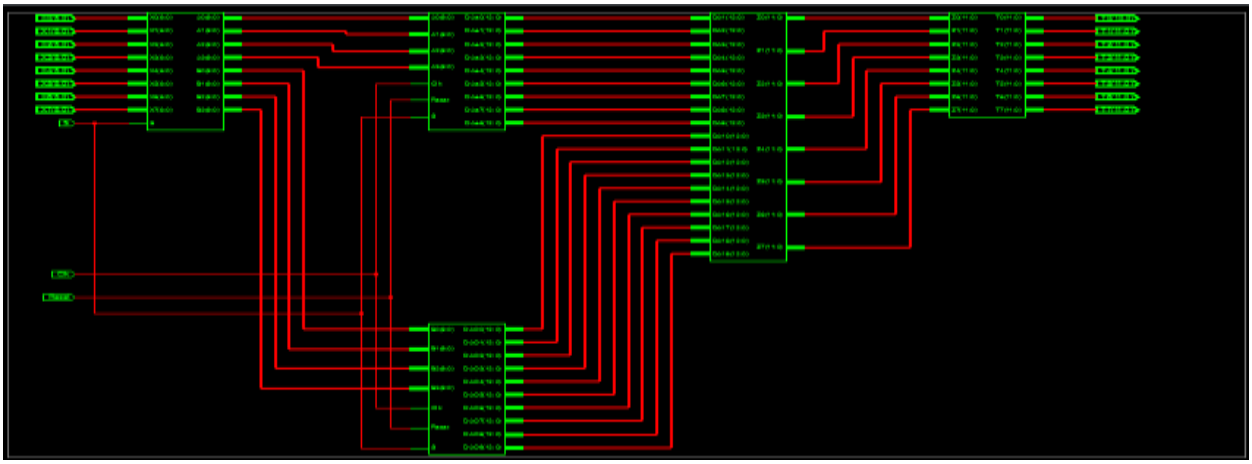

Fig: 27: Schematic view for S_1D_CSDA_MST_1 & S_CSDA_MST_2

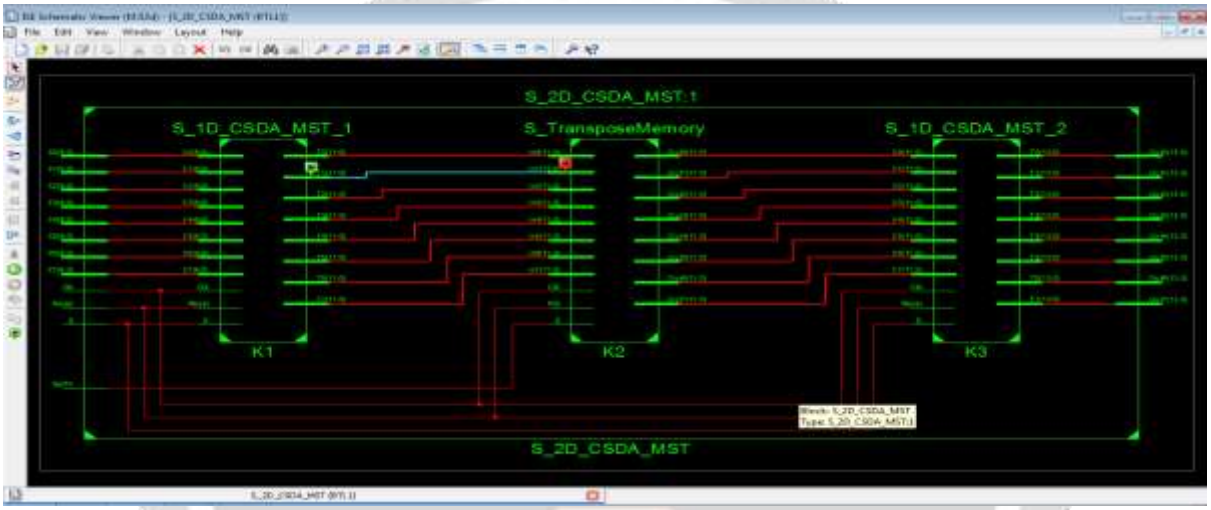

Fig:28: Xilinx Schematic view for S_2D_CSDA_MST

4.TECHNOLOGY SCHEMATIC VIEW:

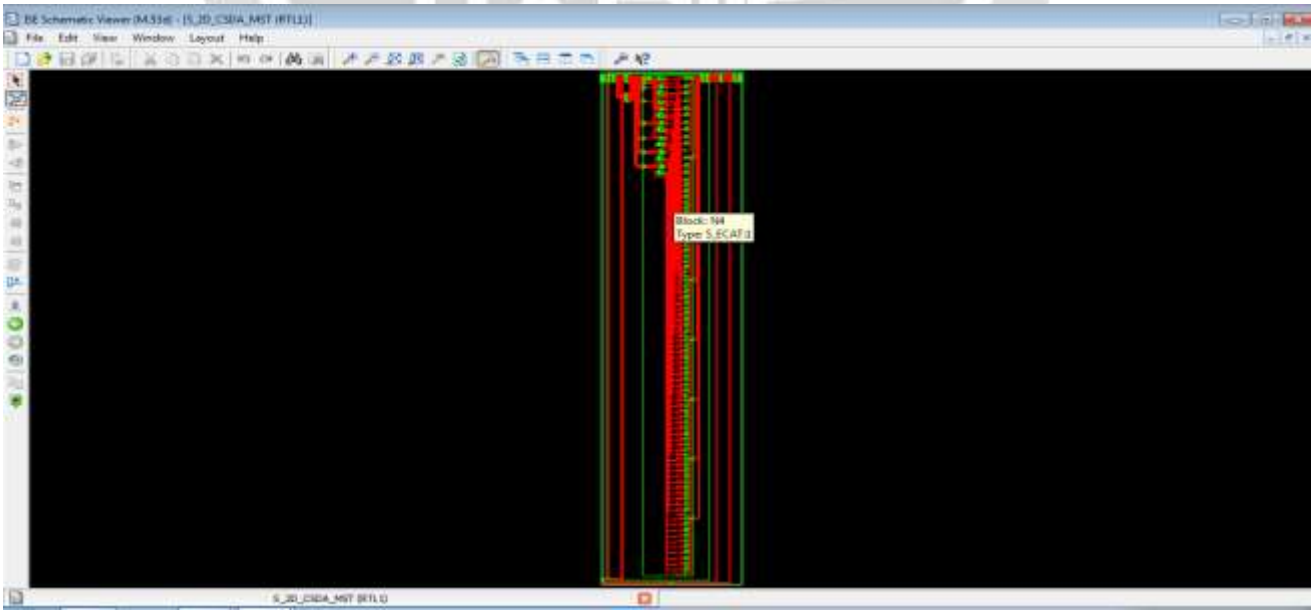

Fig: 29: ISE Schematic View (M.53d)-[S_2d_CSDA_MST (RTL1)]

5. 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.

4. 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, 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.