

Research Paper

POWER AND AREA OPTIMAL IMPLEMENTATION OF 2D-CSDA FOR MULTI STANDARD CORE

Thammisetty Arun Kumar^{1*} and N Suresh Babu²*Corresponding Author: **Thammisetty Arun Kumar** ✉ arun.vyc@gmail.com

This paper proposes a low-cost high-throughput Multi-standard transform (MST) core, which can support MPEG-1/2/4 (8x8), H.264 (8x8, 4x4), and VC-1 (8x8, 8x4, 4x8, 4x4) 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. This achieves a 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. Measurements show that the proposed CSDAMST core achieves a high-throughput rate of 1.28 G-pels/s, supporting the (4928x2048@24 Hz) digital cinema or ultrahigh resolution format. This is possible only with 30k gate counts when implemented in a TSMC 0.18- μ m CMOS process. The CSDA-MST core thus achieves a high-throughput rate supporting Multi-standard transformations at low cost.

Keywords: Common sharing distributed arithmetic (CSDA), discrete cosine transform (DCT), Integer transform, Multi-standard transform (MST)

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 applications.

Numerous researchers have worked on transform core designs, including discrete cosine transform (DCT) and integer transform, using distributed arithmetic (DA) [2]–[9], factor sharing (FS) (Yuan-Ho Chen *et al.*, 2014), 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 (Uramoto S *et al.*, 1992). Yu and Swartzlander

¹ M.Tech. Student, Department of ECE, Chirala Engineering College, Chirala 523155, Prakasam Dt., AP.

² Associate Professor, Department of ECE, Chirala Engineering College, Chirala 523155, Prakasam Dt., AP.

(2001) 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 A M *et al.*, 2006; Peng C *et al.*, 2007; Huang C Y *et al.*, 2008; Chen Y H *et al.*, 2011; Chen Y H *et al.* xxxx; Lai Y K and Lai Y F, 2010). Shams *et al.* (2006) 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 (Peng C *et al.*, 2007; Huang C Y *et al.*, 2008; Chen Y H *et al.*, 2011; Chen Y H *et al.* xxxx).

Chang *et al.* (2009) use 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 (Lee S and Cho K, 2007; Lee S and Cho K (2008). To further reduce the area, present optimization strategies for FS and adder sharing (AS) for multi-standard (MST) applications 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 AP, 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), H.264 and Windows Media Video 9 (WMV-9/VC-1), many MST cores are presented in (Huang *et al.*, 2008), (Lai Y K and Lai Y F, 2010) and introduce a fully supported transform core for the H.264 standard, including 8x8 and 4x4 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 8x8, 8x4, 4x8, and 4x4 transform cores are shown for VC-1, whereas in [9] MST cores supporting the MPEG-1/2/4, H.264, and VC-1 Standards are addressed.

This paper proposes a MST core that supports MPEG-1/2/4 (8x8), H.264 (8x8, 4x4), and VC-1 (8x8, 8x4, 4x8, 4x4) transforms. The proposed MST core employs DA and FS 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- μ m CMOS process, the proposed CSDA-MST core has a throughput rate of 1.28 giga pixels/s (G-pels/s) at 160 MHz. The rate of 1.28 G-pels/s meets the (4928 x 2048@24 Hz) digital cinema or ultrahigh resolution format.

DISTRIBUTED ARITHMETIC

The discrete cosine transform is the basis for the JPEG compression standard. For JPEG, this allows for efficient compression by allowing quantization on elements that is less sensitive. The DCT algorithm is completely reversible making this useful for both loss less and lossy compression techniques.

The DCT is a special case of the well-known Fourier transform. Essentially the Fourier transform in theory can represent a given input signal with a series of sine and cosine terms. The discrete cosine transform is a special case of the Fourier transform in which the sine components are eliminated. For JPEG, a two-dimensional DCT algorithm is used which is essentially the one-dimensional version evaluated twice. By this property there are numerous ways to efficiently implement the software or hardware based DCT module. The DCT is operated two dimensionally taking into account 8 by 8 blocks of pixels. The resulting data set is an 8 by 8 block of frequency space components, the coefficients scaling the series cosine terms, known as basis functions. The First element at row 0 and column 0, is known as the DC term, the average frequency value of the entire block. The other 63 terms are AC components, which represent the spatial frequencies that compose the input pixel block, by scaling the cosine terms within the series.

There are two useful products of the DCT algorithm. First it has the ability to concentrate image energy into a small number of coefficients. Second, it minimizes the interdependencies between coefficients. These two points essentially state why this form of transform is used for the standard JPEG compression technique. By compacting the energy within an image, more coefficients are

left to be quantized coarsely, impacting compression positively, but not losing quality in the resulting image after decompression. Taking away inter-pixel relations allows quantization to be non-linear, also affecting quantization positively. DCT has been effective in producing great pictures at low bit rates and is fairly easy to implement with fast hardware based algorithms.

An orthogonal transform such as the DCT has the good property that the inverse DCT can take its frequency coefficients back to the spatial domain at no loss. However, implementations can be lossy due to bit limitations and especially apparent in those algorithms in hardware. The DCT does win in terms of computational complexity as there are numerous studies that have been completed in different techniques for evaluating the DCT.

The discrete cosine transform is actually more efficient in reconstructing a given number of samples, as compared to a Fourier transform. By using the property of orthogonality of cosine, as opposed to sine, a signal can be periodically reconstructed based on a fewer number of samples. Any sine based transform is not orthogonal, and would have to take Fourier transforms of more numbers of samples to approximate a sequence of samples as a periodic signal.

As the signal we are sampling, the given image, there is actually no real periodicity. If the image is run through a Fourier transform, the sine terms can actually incur large changes in amplitude for the signal, due to sine not being orthogonal. DCT will avoid this by not carrying this information to represent the changes. In the case of JPEG, a two-dimensional DCT is used, which correlates the

image with 64 basis functions.

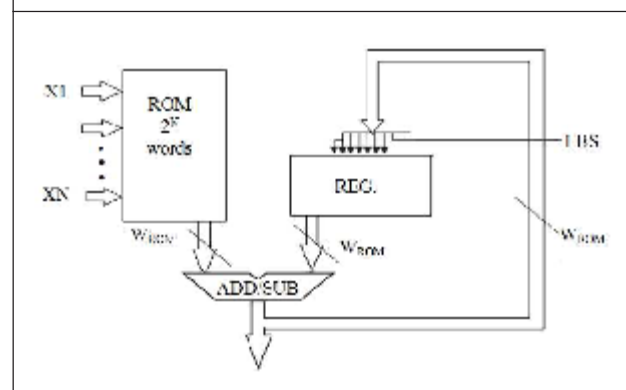
To reduce the truncation errors, multiple error compensation bias methods have been adapted based on the relationship between the partial products and the multiplier and multiplicand. The elements of the truncation part explained in this study are independent. Hence previously described compensation methods cannot be applied. These results in an DA based Error Compensated Adder Tree (ECAT). The ECAT operates shifting and addition in parallel by unrolling all the words to be computed. The ECAT circuit removes the truncation error for high accuracy.

Distributed Arithmetic is highly an efficient technique for computing inner products between a fixed and a variable data. Croisier and Peled designed this method. Liu presented a similar method. The Equation for DA can be written as:

$$Y = a^T X = \sum_{i=1}^N a_i X_i$$

where, $a = 1, 2, 3, \dots, N$ are the coefficients that are fixed. The scaled two's complement representation is used for the data components. Inputs are shifted bit serially out from the shift registers initialize with the least significant bit. Bits are used to be address for the ROM storing Look up Table. The word length in the ROM, W_{ROM} depends on the value of the magnitude and the coefficient of the word length. The parallel implementation of ROM Based Logic is shown in Figure 1. The Inner products that contains many terms can be partitioned into a number of smaller inner products that can be computed and summed as per distributed arithmetic or by using an adder tree. Distributed Arithmetic can also use two or more bits concurrently. Here

Figure 1: ROM Based Logic

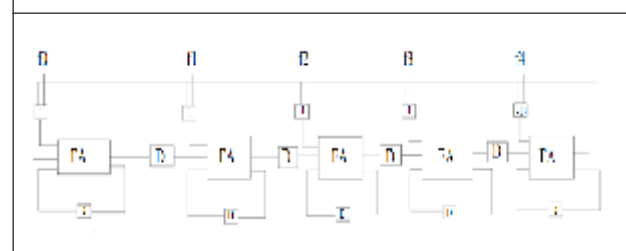


the distributed arithmetic is chosen as key to reduce the area DCT implementation with distributed arithmetic.

Shift Accumulator Logic

The Shift Accumulator Logic is shown in Figure 2. For F_0 , the clock cycle corresponds to the signed bit of data, F_0 should be subtracted. This is performed by adding $-F_0$ and inverting all the bits in F_0 using the XOR gates. After F_0 has been added, the most significant part of the inner product is shifted out of the register accumulator. This can be done by accumulating the Zeros. The clock cycle corresponding the inner ROM is $WD+W_{ROM}$. The carry save adders in the accumulator is let free efficiently by loading the sum and carry bits of the carry save adder into two shift registers. Then the out puts from them are again added by a single carry save adder as shown

Figure 2: Shift and Add Logic



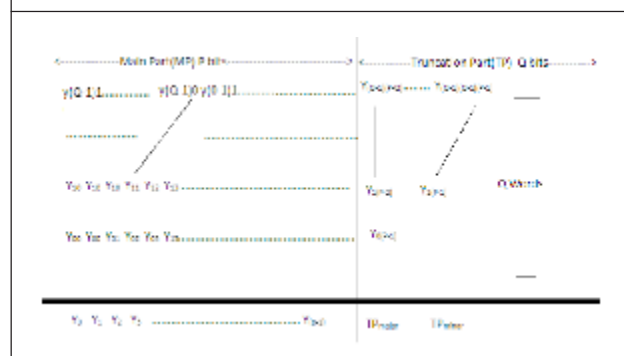
in the Figure 1. This way of computing helps to achieve the double throughput but still the delay and gate counts are major constraints.

Error Compensation

Generally the shifting and addition computation uses a shift and add operator for reduction of hardware in VLSI technology. The computation increases when the number of the shifting and addition increases. Thus the Shift Adder Tree operates by shifting and adding in parallel. A large truncation error occurs in SAT and hence ECAT architecture is proposed in this study to reduce errors. In the Figure 3 shown below the QP bit word operates the shifting and addition in parallel. The operation can be divided into two parts, The Main Part, that contains the Most Significant Bits and the Truncation Part that consists of the Least Significant Bits. The shifting and addition output can be written as: The output Y will gain the P-bit MSBs applying a rounding operation called post truncation which is represented as (Post-T). The hardware cost increases in the VLSI design anyway. Commonly the TP is truncated to reduce the hardware cost in parallel shifting and adding operations known as the Direct Truncation method. A large truncation error occurs due to the negation of carry propagation from the truncation part to most significant part. In order to remove the truncation errors many error compensation bias methods have been presented. All the efforts result in a fixed width multiplier. The products in the multiplier have a relationship between the input multiplier and the multiplicand.

This type of compensation method uses the correlation of inputs to calculate a fixed or an adaptive compensation bias using simulation

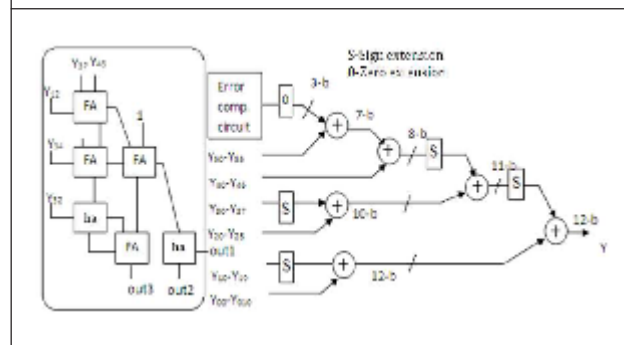
Figure 3: Error Compensation Technique



or the statistical analysis. The error compensation by shifting and adding operation is given in the following Figure 4. The figure shows about the sequence of shifting and additions. It consists of sign extension bits and zero extension bits. Every sign extension is represented with 'S' and zero extension bit with '0'. The bits are serially shifted and then added with the next four coming bits from the registers. The comparisons of the Absolute average error, the Maximum error and the mean square error will be discussed in the simulation environment.

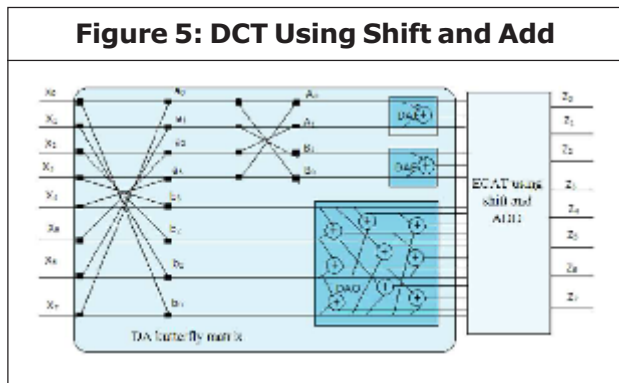
The adders reduce the chip area of DCT core. A speed limitation occurs in the shift and adds multiplier. The following Figure 5 shows the implementation of the DCT using an ECAT with shift and add logic. The error

Figure 4: Error Compensation Using Shift and Add Logic



compensation is performed at the end of the DCT. The DCT is freed of error at its output with the help of an ECAT with Shift and Add Logic.

BASELINE 2-D DCT ARCHITECTURE



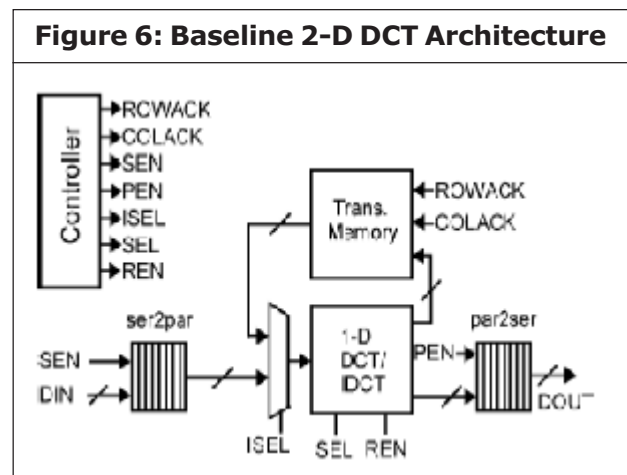
The Baseline 2-D DCT Architecture provides a reference design for application of our low power techniques. It is based on Chen Algorithm. This acts as a reference design for the computation of power savings techniques. This approach requires three steps: eight 1-D DCT/IDCTs along the rows, a memory transposition, and another eight 1-D DCT/IDCTs along the transposed columns. A block diagram of the Baseline Architecture shown above includes the controller which enables input of the first row of data (DIN) through the ser2par unit under the SEN signal. It then activates the 1-D DCT unit with the SEL and REN signals determining the data path. The first row of the transposition memory stores the results & the process repeats for the remaining seven rows of the input block. Next, the ISEL and COLACK signals enable the 1-D DCT unit to receive the input data from the columns of the transposition memory. The final results of the column-wise 1-D DCT are

available at the output.

Fully Pipelined Architecture

In this architecture, a row output vector is computed using multipliers, multiplexers, accumulators, and registers. The elements of input vector X are fed into the circuit one at a time. The 8 output elements are computed simultaneously and are shifted out serially. An input vector is multiplied by the coefficient matrix M' to get the output. The second element of Y is computed through some additions and subtractions of the elements of X, and then multiplied by constant a. Permutation is done before the result goes into the accumulator.

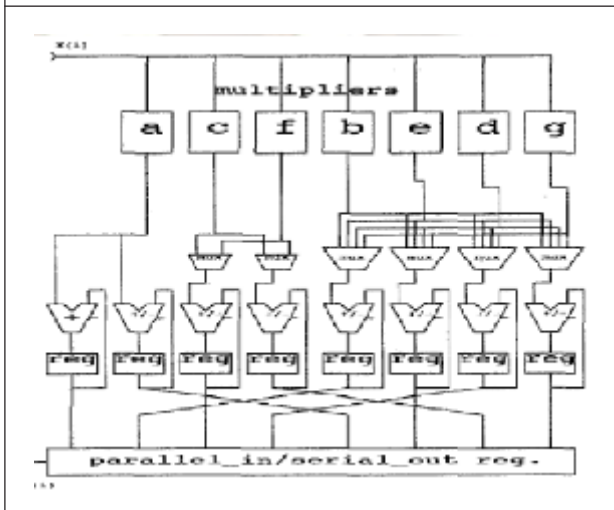
The circuit accepts one pixel per clock cycle



and the entire processing is performed as a linear pipe. When the left column of register set RS is filled with eight data elements, the entire column is copied onto the corresponding registers in the right column. A similar process occurs in each of the partitions simultaneously. The transpose buffer consists of an 8x8 array of register pairs, the data is input to the transpose buffer in row-wise fashion until all the 64 registers are loaded.

The data in those registers are copied in

Figure 7: Fully Pipelined 2-D DCT Architecture

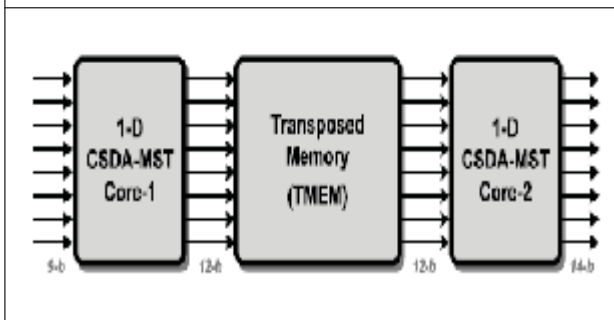


parallel onto the corresponding adjacent registers which are connected in column-wise fashion. While the data is being read out from the column registers, the row registers will keep receiving further data from the DCT module. Thus, the output of row-wise DCT computation is transposed for column-wise DCT computation.

DISCUSSION AND COMPARISONS

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.

Figure 8: Proposed 2-D CSDA-MST Core

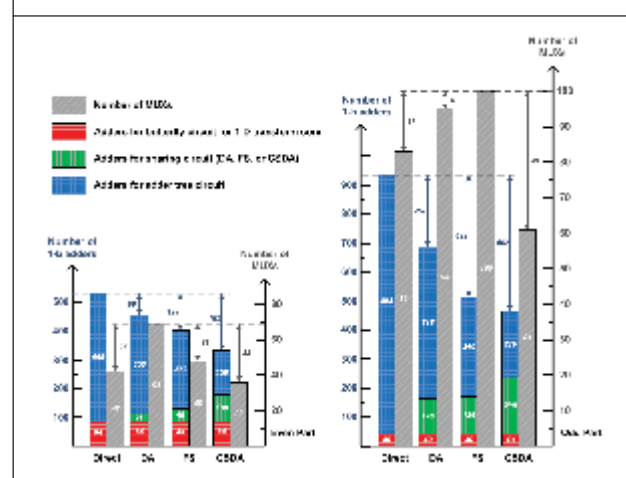


A. Hardware Resources Evaluation for Proposed CSDA

Method

Figure 9 shows a summary of the usage for the adders and MUXs 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 second and third group bars, respectively, denote that only DA and FS strategies are adopted, and the last group bars show the proposed CSDA method applied to the 1-D MST core. The distribution for red, green, blue, and gray blocks indicates the number of adders for the butterfly circuit, sharing circuit (including DA, FS, and CSDA), adder tree, and number of MUXs, respectively. 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;

Figure 9: Number of 1-bit adders and MUXs for the CSDA Method as Applied to 1-D MST Core

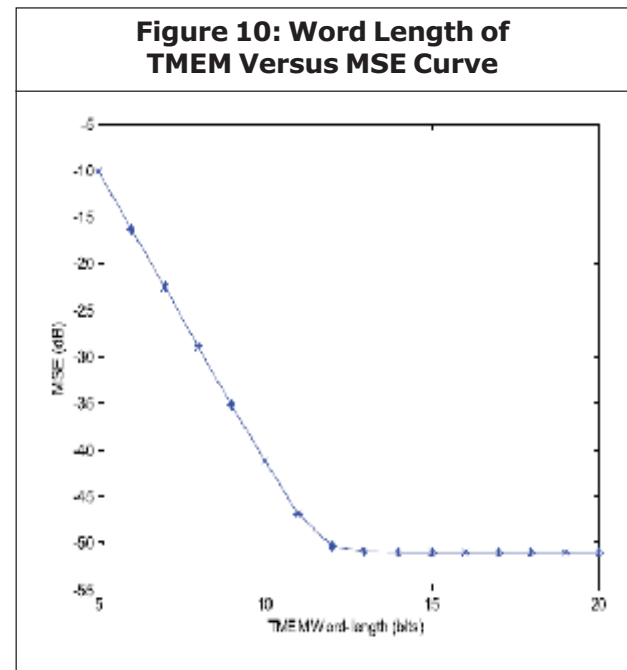


326(= 69 + 257) and 551(= 129 + 422) 1-bit adders are reduced for these methods, improving the adder cost when compared with direct implementation. Fortunately, the proposed CSDA method can achieve a saving of 655(= 193 + 462) adders when compared with direct implementation. It can further use fast adder 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.

B. Accuracy Analysis for CSDA-MST Core

The specific foreman sequence is used to verify system accuracy in this paper. After inputting the original test image pixels to the proposed 2-D CSDA-MST core, the transform output data are captured and fed into MATLAB to compute the inverse DCT by using 64-bit double-precision operations. The average peak signal-to-noise-ratio (PSNR) of each standard among different dimensions. Generally, the human eye cannot recognize differences between images when the PSNR values are larger than 40 dB. Therefore, all the reconstructed images using the CSDA-MST core exhibit excellent subjective quality because of their high PSNR values (larger than 45 dB). Furthermore, the proposed

CSDAMST core also does the word length of TMEM versus mean square error (MSE) analysis, and the results are established as a curve in Figure 10. The appropriate word length of TMEM is 12 bits, which can be obtained from the curve of Figure 10.



C. Comparisons with Other Transform Architectures

A comparison of the proposed 2-D CSDAMST core with previous works. In the DA-based row-column decomposition methods (Huang C Y et al., 2008), eight parallel computation paths cause the transform core to achieve a throughput rate of 400 M-pels/s when operated at 50-MHz operation frequency. The scalable-DA 1-D transform core supporting MPEG-1/2/4 8x8 and H.264 4x4 transforms is presented. The core supports MPEG-1/2/4 (8x8), H.264 (8x8, 4x4), and VC-1 (8x8, 8x4, 4x8, 4x4) multiple transforms by using the DA method to achieve a high throughput rate of 800 M-pels/s, as implemented in a 90-

nm process (Lai Y K and Lai Y F, 2010). In the direct 2-D method is presented to implement the 2-D transform core. The high-performance core is presented by expanding the 4x4 transform to eight pixels per cycle. However, the 2-D core supports only the 4x4 transform in the H.264 standard permutation matrices to implement the 2-D integer transform core, to support 8x8 and 4x4 transforms for H.264 and VC-1 standards. However, the hardware cost is high because of the multiplier-based structure. The common architecture supports MPEG-1/2/4, H.264, and VC-1 standards by using the common delta circuits (Chang *et al.*, 2009), (Lee S and Cho K, 2007). The small throughput rate is caused by taking a large number of cycles for calculating each macro block (MB) present a high-throughput and cost-effective architecture in H.264 coders by using permutation and matrix factorization. This approach applies the addition-shifting operation to implement the multiplication operation. However, part of the architecture is idle when the system operates in the four-point transform. Therefore, the throughput rate of the four-point transform is half that of the eight-point transform. Furthermore, the architecture cannot perform MPEG-1/2/4 and VC-1 transforms introduce the 2-D 4x4, 4x8, 8x4 and 8x8 VC-1 transform core by using hardware sharing architecture. The DCT circuit for MPEG-1/2/4 is larger than that of other compression standards in implementing these standards. Unfortunately, architectures only support the H.264 compression standard to achieve higher hardware efficiency than the proposed MST, where the hardware efficiency is defined as follows:

$$\text{Hardware Efficiency (} 10^3 \text{ pels / sec- gate - W)} = \frac{\text{Throughput Rate}}{\text{Power} \times \text{Gate Counts}}$$

The power consumption is another issue in the circuit designs. In order to have a fair comparison, the power efficiency is defined as throughput rate per power and gate. The proposed CSDA-MST core has medium power consumption between pervious works, and the power efficiency is

$$\text{Power Efficiency (pels / sec- gate - W)} = \frac{\text{Throughput Rate}}{\text{Power} \times \text{Gate Counts}}$$

The proposed CSDA-MST core can support MPEG-1/2/4 (8x8), H.264 (8x8, 4x4), and VC-1 (8x8, 8x4, 4x8, 4x4) multiple transforms by using the CSDA algorithm to reduce area cost. Furthermore, by employing eight parallel computation paths, the CSDA-MST core can achieve a high throughput rate of 1.28 G-pels/s, and the four-point transform can achieve the same throughput rate because the CSDA_O can execute four-point transform. Therefore, the CSDA-MST core has high hardware efficiency when supporting MPEG-1/2/4, H.264, and VC-1 transformations.

D. Chip Implementations

To verify the proposed CSDA-MST core, the RTL hardware design of the proposed core is synthesized using the Artisan TSMC 0.18- μ m Process Standard Cell and the Synopsys Design Compiler tool, whereas the Cadence SoC Encounter tool is applied for placement and route (P&R). The priority of design constraint is speed, and then area. The delay

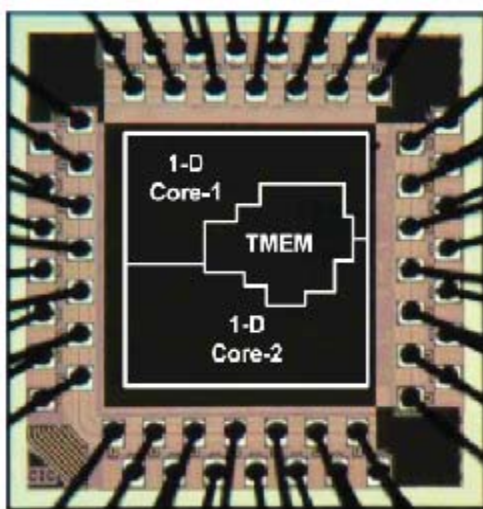
constraint is set as 6 ns, and the core area is set as small as possible. Thus, the proposed core can achieve a small area cost and hold on an operating frequency of 160 MHz. The proposed design is synthesized and physically implemented in a 1.8-V TSMC 0.18- μ m CMOS process by TSMC company. Measured results show that the CSDA-MST core has an operating frequency of 160 MHz. Furthermore, the proposed CSDA-MST core can use the same design strategy and reduce the pipeline stage to meet the design constraint by reducing area cost when implemented in the latest technologies, such as 90-, 45-, 32-nm technologies, etc. Because of eight parallel computation paths, the proposed CSDA-MST core can achieve a throughput rate of 1.28 G-pels/s, which is higher than the specifications of digital cinema (4928 x 2048@24 Hz) with 4:4:4 video format. Figure 11 shows a photomicrograph of the proposed core with each module's boundary. For future extra-high-resolution standards, a 1.98-V

supply voltage can be applied to the fast mode. The 1.64 G-pels/s (= 8 x 205 MHz) high-throughput rate is achieved for the proposed CSDA-MST core with 109 mW power consumption.

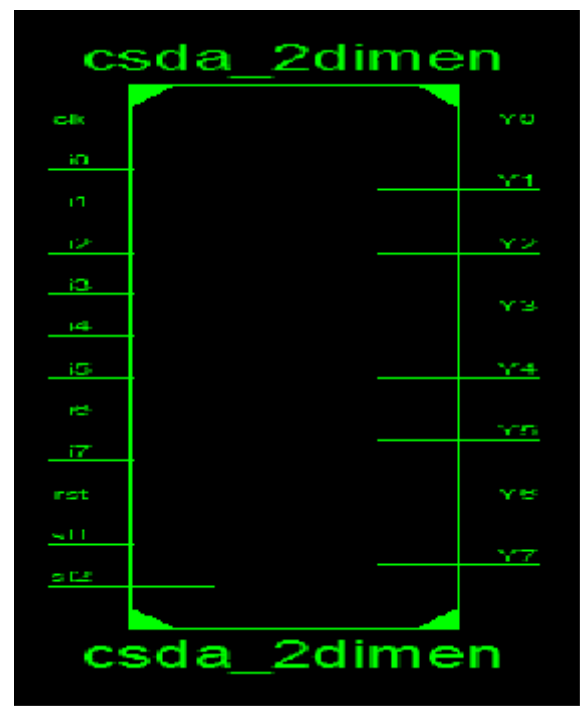
E. Forward and Inverse Transform Implementations

To demonstrate the proposed CSDA-MST applying to the video decoder, the proposed CSDA-MST core is extended to support the inverse transform in multiple standard applications. The Synopsys Design Compiler is applied with an Artisan TSMC 0.18- μ m standard cell library to implement the selection forward and inverse CSDA-MST core, and the Prime Time PX is used to estimate the power consumption. The core layout and simulated characteristics are shown in Figure 12. Operated at 160 MHz clock frequency, the proposed core can achieve.

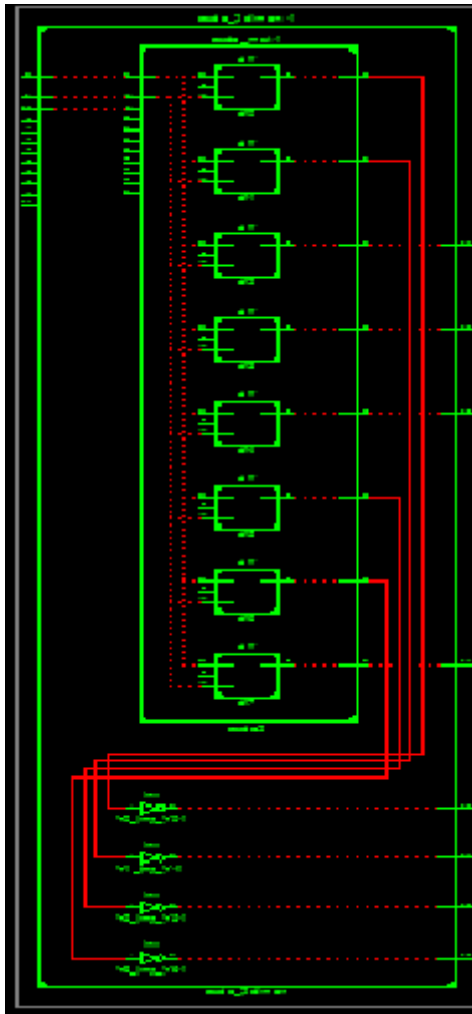
Figure 11: Photomicrograph of the Proposed CSDA-MST Core



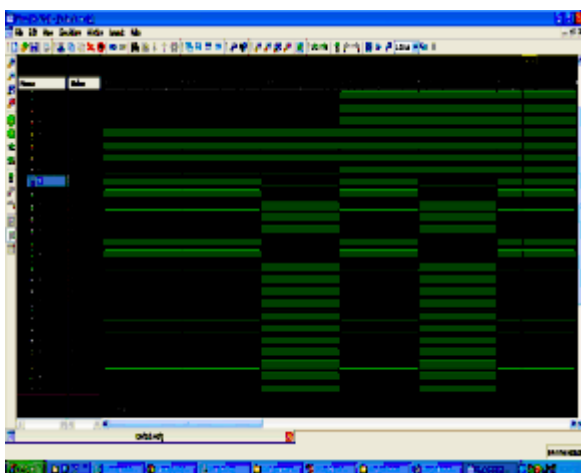
RTL Schematic



Internal View of a Design



Wave form of CSDA



CONCLUSION

The CSDA-MST core can achieve high performance, with a high throughput rate and low-cost VLSI design, up porting 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. Chang H, Kim S, Lee S, and Cho K (2009), "Design of area-efficient unified transform circuit for multi-standard video decoder," in *Proc. IEEE Int. SoC Design Conf.*, Nov., pp. 369–372.
2. Chen Y H, Chang T Y and Li C Y, "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.
3. Chen Y H, Chang T Y, and Li C Y (2011), "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.
4. Huang C Y, Chen L F and Lai Y K (2008), "A high-speed 2-D transform architecture with unique kernel for multi-standard video applications," in *Proc. IEEE Int. Symp. Circuits Syst.*, May, pp. 21–24.

5. Lai Y K and Lai Y F (2010), "A reconfigurable IDCT architecture for universal video decoders," *IEEE Trans. Consum. Electron*, Vol. 56, No. 3, pp. 1872–1879.
6. Lee S and Cho K (2008), "Architecture of transform circuit for video decoder supporting multiple standards," *Electron. Lett*, Vol. 44, No. 4, pp. 274–275.
7. Lee S and Cho K (2007), "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., pp. 102–107.
8. Peng C, Cao X, Yu D, and Zhang X (2007), "A 250 MHz optimized distributed architecture of 2D 8x8 DCT," in *Proc. 7th Int. Conf. ASIC*, pp. 189–192.
9. Shams A M, Chidanandan A, Pan W, and Bayoumi M A (2006), "NEDA: A low-power high-performance DCT architecture," *IEEE Trans. Signal Process.*, Vol. 54, No. 3, pp. 955–964.
10. Uramoto S, Inoue Y, Takabatake A, Takeda J, Yamashita Y, Terane H, and Yoshimoto M (1992), "A 100-MHz 2-D discrete cosine transform core processor," *IEEE J. Solid-State Circuits*, Vol. 27, No. 4, pp. 492–499.
11. Yu S and Swartzlander E E (2001), "DCT implementation with distributed arithmetic," *IEEE Trans. Comput.*, Vol. 50, No. 9, pp. 985–991.
12. Yuan-Ho Chen, Jyun-Neng Chen, Tsin-Yuan Chang, and Chih-Wen Lu (2014), "High-Throughput Multi-standard Transform Core Supporting MPEG/H.264/VC-1 Using Common Sharing Distributed Arithmetic", *IEEE Transactions on Very Large Scale Integration Systems*.