Data Dependency Considerations in Low Power Design of Discrete Cosine Transform Architecture

Satyanadh Gundimada, and Vijayan K. Asari
VLSI Systems Laboratory
Department of Electrical and Computer Engineering
Old Dominion University, Norfolk, VA 23529, USA

Abstract

A low power design approach for the implementation of Discrete Cosine Transform coefficients is presented in this paper. The architecture takes into consideration that the neighboring elements of a vector in image and video data are closer to each other in magnitude. The elements of the fixed coefficient matrix in the DCT computations are distributed such that the architecture doesn't need any multipliers. By introducing the concept of even and odd group DCT coefficient computations into the proposed Distributed Arithmetic based architecture, a reduction of 91.56 % in the number of adders has been achieved over the direct implementation. A circuitry to control the switching activity of the dynamic adders based on the data dependency in the input sequence is designed which will keep some of the adders non functional when similar inputs are presented. Evaluation of the new concept for the design of an 8 x 8 DCT coefficient computation module shows considerable reduction in power and about 35.7% reduction in VLSI area.

Key words— Distributed arithmetic, discrete cosine transform, data dependency, low power design, switching activity, data dependency.

1. Introduction

Discrete Cosine Transform (DCT) based image coding is the basis for all the image and video compression standards. With portable multimedia applications on the rise in the recent years and the trends showing that the rise will continue, power consumption of VLSI chips has become an important factor. Since the battery technology rise is not on par with the demands of power for portable electronic applications, it becomes essential that other means of power savings in VLSI systems should be approached. There are various methods in which reduction can be achieved starting from architectural level designs for power reductions to device level designs. This paper tries to incorporate the concept of reducing switching capacitance power at the nodes, which is the basic component of power consumption in dynamic VLSI circuits. Discrete Cosine Transform is one of the widely researched fields over the past decade. Various algorithms [1] and architectures [13-17] were proposed and implemented for various advantages including area, and power [11],[12]. Distributed arithmetic (DA) found its way into many of the architectures of DCT [6-10]. The most widely implemented distribute arithmetic based DCT designs make use of ROM. The coefficients are pre-computed and are stored inside the ROM. The ROM addresses are accessed based on the input data. Other methods use Multiply Accumulators (MAC) as the basic component to implement DCT. Xanthopolous and Chandrakasan proposed an adaptive bitwidth and arithmetic activity based low power DCT architecture, which takes into consideration, signal correlations and quantization [2]. NEDA or the "New Distributed Architecture," [3] unlike the conventional DA where in the input bits are distributed, is implemented with the distributed coefficients. NEDA architecture is one of the designs which are not based on ROM. Though the ROM based architectures are much simpler to realize, the serial execution of those architectures makes them slower when the input data precision increases and makes them not so suitable for high performance applications.

DCT is the main component in video codecs and is the major power consuming unit. This has led to the design of the low power architecture proposed in this paper. Neighboring pixel magnitudes in image and video data are closer to each other. The researched statistics show that the neighboring data do not have much difference in their most significant bits. Hence taking into consideration, the idea of data closeness, a new architecture has been built to suite low power applications using DCT.

A scenario of 8 x 8, 1-D DCT is explained. It uses simple mathematical equations, and can be realized using only stages of adders and shifters. Control circuitry is provided for the adder arrays to keep the adders idle during the addition of similar data. Data Compression achieved by the design and the power management measures are also explained. Initial results indicate that compression is achieved in terms of number of adders and with reduced power. The main advantage of the design is its compactness and simpler control circuitry to achieve low power design.

The paper is organized as follows: Functional and mathematical description of the proposed method is described in the section 2. Section 3 gives a detailed description of the proposed architecture. The method followed for reducing the number of adders in the adder array taking into consideration the redundant rows of the coefficient matrix is given in section 4. A control circuitry description for the adder arrays is given in section 5. The results obtained for the proposed architecture and comparisons with other published results are given in the section 6. Section 7 provides a conclusion.

2. Functional and mathematical description

2.1. One - dimensional DCT

An 8 x 8 1-D DCT is realized using the proposed architecture and could be extended to a 2-D DCT, with a transposition memory in between the two 1-D DCTs. The basic one-dimensional DCT equation is given as.

$$Y_i = \frac{c(i)}{2} \sum_{j=0}^{7} X_j \cos\left(\frac{(2j+1)i\pi}{16}\right)$$

$$\text{where } c(i) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for i=0} \\ 1, & \text{otherwise} \end{cases}$$

2.2. Chen's algorithm

The first step in realizing the architecture for the 1-D DCT is to reduce the 8 x 8 coefficient matrix into two 4x4 matrices using Chen's algorithm [5]. This step divides the computations of DCT coefficients into even and odd parts and the process can be used to the advantage of realizing the low power design. It also achieves the adder array compression.

$$\begin{bmatrix} Y_0 \\ Y_2 \\ Y_4 \\ Y_6 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} A & A & A & A \\ B & C & -C & -B \\ A & -A & -A & A \\ C & -B & B & -C \end{bmatrix} \begin{bmatrix} X_0+X_7 \\ X_1+X_6 \\ X_2+X_5 \\ X_3+X_4 \end{bmatrix}$$

$$\begin{bmatrix} Y_1 \\ Y_3 \\ Y_5 \\ Y_7 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} D & E & F & G \\ E & -G & -D & -F \\ F & -D & -G & E \\ G & -F & E & -D \end{bmatrix} \begin{bmatrix} X_0-X_7 \\ X_1-X_6 \\ X_2-X_5 \\ X_3-X_4 \end{bmatrix}$$

The 8 input pixel values for the 1-D DCT are provided in 2's complement form. In the above matrix equations, X_0 to X_7 are the inputs and Y_0 to Y_7 are the output DCT coefficients. A, B, C, D, E, F, G are the fixed coefficients and they are defined as: A = cos (π/4), B = cos (π/8), C = sin (π/8), D = cos (π/16), E = cos (3π/16), F= sin (3π/16), G = sin (π/16)

2.3. Constant coefficient distribution

By providing the step of even and odd coefficient division, one can expect a reduction in the complexity of the circuit. The ½ in the equations can be taken inside the matrix. The idea is to separate out the computations and take advantage of the data redundancy. The fixed coefficients are also expressed in 2's complement form. For any equation of type

$$Y = \sum_{l=0}^{i-1} P_l R_l \qquad (1)$$

where P_l are fixed values and R_l are varying values.

P_l can be expressed as

$$P_l = -P_l^M + \sum_{j=N}^{M-1} P_l^j \, 2^j \qquad (2)$$

P_l^j = 0 or 1; j=N, N+1....M-1, and P_l^N is the LSB and P_l^M is the most significant bit and it is the sign bit. P_l can be distributed as in [3]. For the case of i being 4, Y can be expressed in matrix notation as.

$$Y = \begin{bmatrix} P_0 & P_1 & P_2 & P_3 \end{bmatrix} \begin{bmatrix} R_0 \\ R_1 \\ R_2 \\ R_3 \end{bmatrix} \quad (3)$$

Eqn (3) can be reduced to eqn (4) by expressing the fixed values into 2's complement form.

$$Y = \begin{bmatrix} 2^N 2^{N+1} \ldots 2^M \end{bmatrix} \begin{bmatrix} P_0^N & P_1^N & P_2^N & P_3^N \\ P_0^N & P_1^N & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ -P_0^N & -P_1^N & -P_2^N & -P_3^N \end{bmatrix} \begin{bmatrix} R_0 \\ R_1 \\ R_2 \\ R_3 \end{bmatrix} \quad (4)$$

$$Y = \begin{bmatrix} 2^N 2^{N+1} \ldots 2^M \end{bmatrix} \begin{bmatrix} 100\cdots\cdots0 \\ 010\cdots\cdots0 \\ \vdots \\ 000\cdots(INV) \end{bmatrix} \begin{bmatrix} P_0^N & P_1^N & P_2^N & P_3^N \\ P_0^{N+1} & P_1^{N+1} & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ P_0^M & P_1^M & P_2^M & P_3^M \end{bmatrix} \begin{bmatrix} R_0 \\ R_1 \\ R_2 \\ R_3 \end{bmatrix} \quad (5)$$

The sparse matrix described in [3] is given in eqn (5) accounts for the 2's complement representation of the binary fixed coefficient matrix. From Y, Y' is can be written as:

$$Y' = \begin{bmatrix} P_0^N & P_1^N & P_2^N & P_3^N \\ P_0^{N+1} & P_1^{N+1} & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ P_0^M & P_1^M & P_2^M & P_3^M \end{bmatrix} \begin{bmatrix} R_0 \\ R_1 \\ R_2 \\ R_3 \end{bmatrix} \quad (6)$$

Y' is a Mx1 matrix and can be realized using an adder network with R_0, R_1, R_2, R_3 as inputs. Y can then be obtained form Y' through a series of shifts and additions with suitable circuitry for sign bit in place.

For example, Y_0, which is a DCT coefficient, can now be expressed as:

$$Y_0 = \begin{bmatrix} A' & A' & A' & A' \end{bmatrix} \begin{bmatrix} X_0 + X_7 \\ X_1 + X_6 \\ X_2 + X_5 \\ X_3 + X_4 \end{bmatrix} \quad (7)$$

Y'_0 matrix corresponding to the Y_0 is derived for a precision of 13 bits for the row matrix in eqn (7). A' is now A/2.

$$Y'_0 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X0 + X7 \\ X1 + X6 \\ X2 + X5 \\ X3 + X4 \end{bmatrix} \quad (8)$$

The Y'_0 matrix can be realized using an adder array. This is explained in the adder compression section - section 4.

3. DCT architecture

Figure1 shows the block diagram of the proposed architecture. X_0 - X_7 are the 8 input elements and Y_0 - Y_7 are the 8 1-D DCT coefficients. The first set of adders shown in the block diagram realizes the Chen's algorithm by performing the division of computations of DCT coefficients into even and odd parts. The input bits are taken to be of 8 bit precision and hence 8 bit adders are used. The next adder array is arranged according to the distributed binary coefficient matrix. The second array has 10 bit adders. The even DCT coefficient additions are divided into two bit slices of 6 and 4 instead of a single adder of 10 bits. The dynamic adders used for the purpose can be disabled using the control signal in accordance with the clock. The 'Same MSB detector' is a control block and it detects if the 6 most significant bits of the inputs to the adder array are same or not. If the inputs to the adder array have the same most significant 6 bits then

a control signal is used to disable all the 6 bit adders within the adder array, where as the 4 bit adders still continue to add as usual. The arrangement of adders into 6 and 4 bits is taken up considering the statistics of the image data [4], which suggests that about 70-80 % of vector elements of the data can be very much closer. The shift and add block has a proper 2's complement circuitry to take care of the sign bits. It is provided with the inputs from the second adder array in full. Equal most significant bits are shifted and appended to the register contents at the input of the shifters when the 6 bit adders remain idle. The impact of the 6 bit adders not working for the time during which the inputs are similar would lead to significant power gains.

The second half of the block diagram is for the odd part of the DCT coefficient generation. One of the characteristics of this data is that it would be small and is represented in two's complement form. The 'Equal MSB compare' block finds out if the seven most significant input bits to the second adder array are all zeros or all ones. The adders in this array are divided into 7 and 3 bits for a 10 bit precision. This is based on the fact that data is small. Once the equal bit comparator compares the inputs to the adder, control signals are sent to the 7 bit adders such that they do not operate and remain idle. The control signals are produced on the basis of addition of most significant zeros with most significant zeros and similarly ones with ones. It does not take care of addition of zeroes with ones. The data now is written back to the registers at the input of the shifters, depending on the type of the control signal.

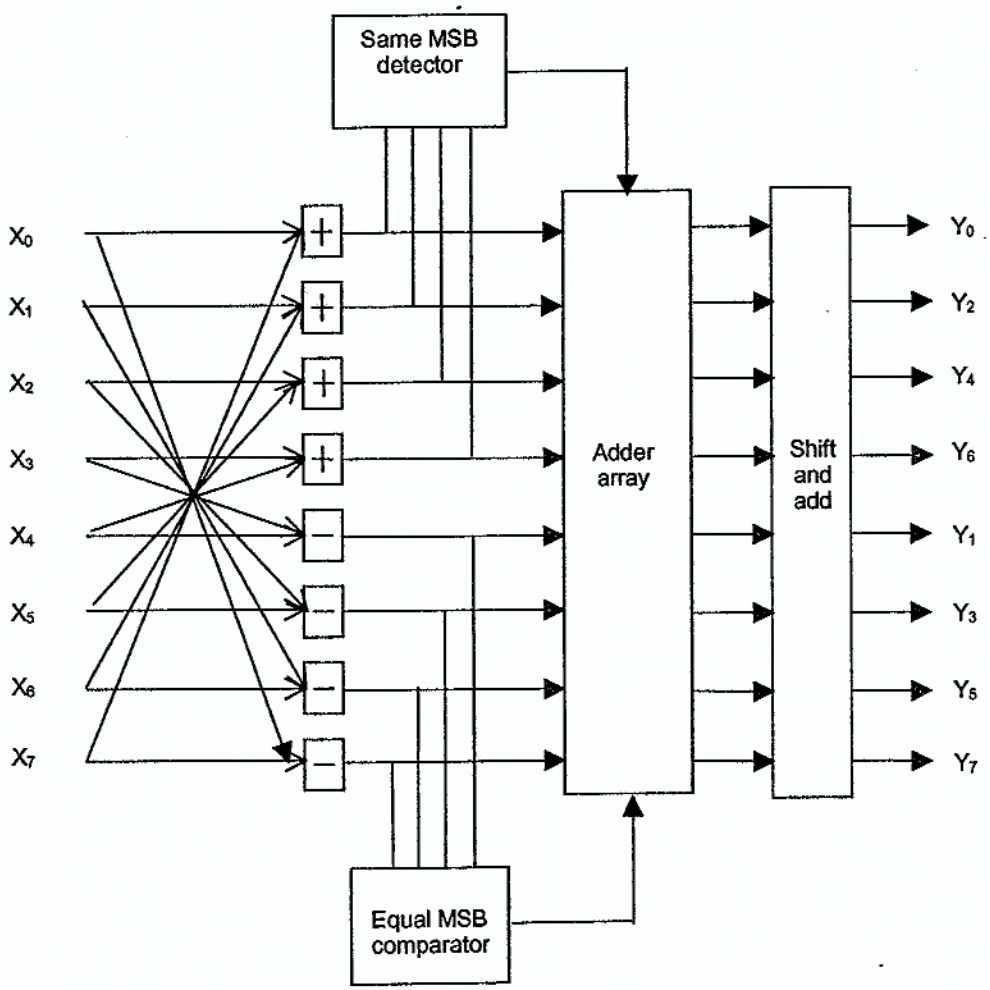

Figure 1. Block diagram depicting the proposed DCT architecture with adder arrays and control blocks.

4. Compression using adders

Adder arrays are realized according to the distribution of coefficient matrix described in section 2.

4.1. Realization of adder array

A distribution of coefficient matrix can be observed in the eqn 2.8. The adder array is realized according to the distribution show in Figure 2.

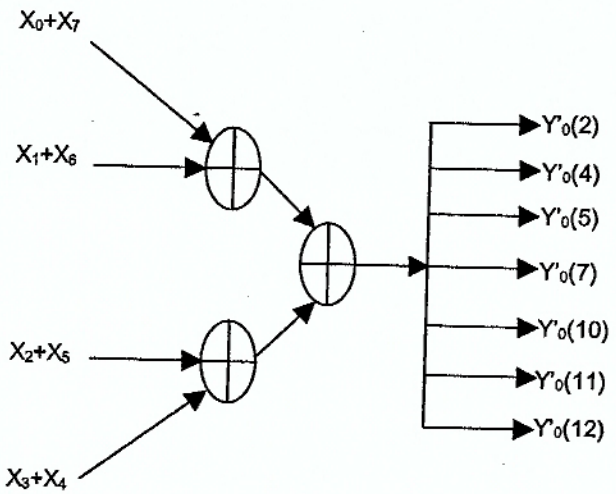

Figure 2. Adder array for realizing Y'_0.

It can be seen that it requires only three 10 bit adders to obtain Y'_0. In the case of the above array, it only requires one summation of X_0 to X_7. Thus the obtained result of the summation can be used to get Y'_0. The rows with the Zeros in the matrix need not be considered since they any how result in a zero summation. Similarly Y'_1, Y'_2 ..... Y'_7 can also be realized by taking into consideration the inherent redundancy within the matrix.

4.2. DCT coefficient generation

The block diagrams shown in Figure 3(a) and Figure 3(b) represent the adder arrays realized for obtaining the odd and even 1-D DCT coefficients respectively. A precision of 13 bits for the coefficients is considered. The adders are arranged according to the binary matrices of the entire even coefficient rows. There exists a lot of redundancy in the rows of the distributed coefficient matrix as explained earlier. Totally 7 adders are required for the generation of 4 even 1-D DCT coefficients. Each adder shown in the figure is in fact a combination of 6 bit and 4 bit adders. It required a total of 11 adders to realize the 4 odd DCT coefficients. Each adder in this case is a combination of 7 bit and 3 bit adders. Whenever the adder array receives a control signal saying that the MSB's are same, the 6 bit adders are disabled and the shifted data is forwarded to the registers inputting to the shifters. The explanation for the working of the proposed control circuitry is given in the next section. The adder array for the generation of the odd DCT coefficients is shown in Fig 3(b). A total number of 26 adders are needed to realize the DCT coefficients for a precision of 13 bits for the fixed cosine coefficients. The blocks SMD and EMC are explained in detail in the next section.

5. Dynamic control of adders

The adders used in realizing the adder arrays are dynamic adders. Hence the adders work on the initiation of clock. This concept of dynamic adders is taken to the advantage of realizing the aim of low power architecture. When a computation by the adder is not needed, the clock is turned off for the adder. A control circuitry for this purpose is realized. There are two parts for the control. One is termed as "Same MSB Detector (SMD)" and the other is called "Equal MSB Comparator (EMC)".

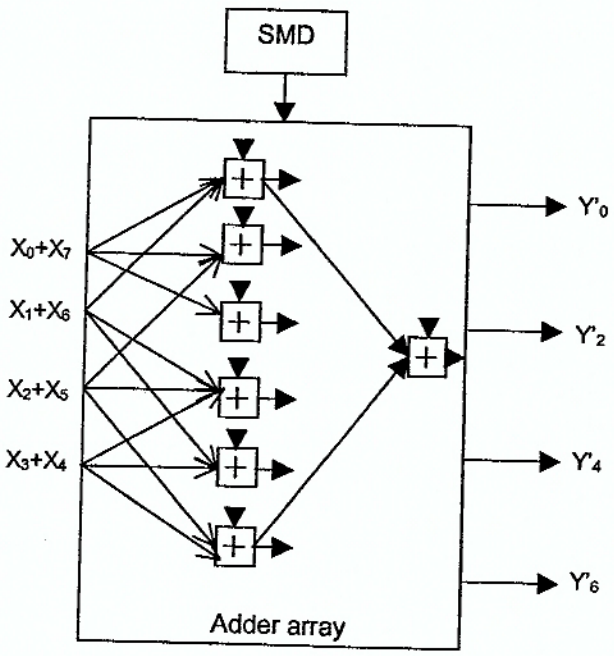

Figure 3 (a). Block diagram depicting the adder array for even DCT coefficient generation along with control signals.

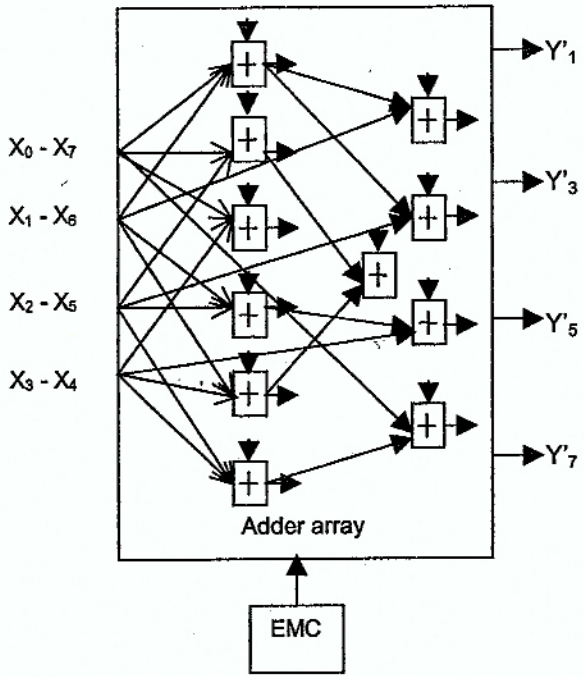

Figure 3(b). Block diagram depicting the adder array for odd DCT coefficient generation along with control signals.

5.1. Same MSB detector

The SMD is designed to produce a control signal to stop the clock to the 6 bit adders within the adder array producing the even DCT coefficients, whenever the most significant 6 bits of the input bits are equal. This will stop those adders from working; in other words, will stop the switching of the capacitive nodes.

5.2 Equal MSB comparator

The EMC on the other hand produces control signal to stop the clock to the 7 bit adders in the array based on whether the most significant 7 bits are all zeroes or all ones. When adders receive the corresponding control signals, their computation is stopped and hence remain idle which would otherwise be unnecessarily computing. This reduces the switching activity in the dynamic adders and would result in saving power. The rest of the adders in the array continue to work normally. The most significant addition which is not computed by the adders can be realized by just shifting those bits by one position to left. The shifted most significant bits are in turn provided at the output along with the computed addition of the remaining least significant bits. The idea of having the sliced adders and control circuitry is driven by the fact that most of the inputs to the arrays are closer in magnitude to each other.

6. Performance evaluation

The proposed architecture is faster when compared to the other ROM based 1-D DCT designs that are available today. Since the ROM based distributed arithmetic realizations work on serial data, they are much slower compared to the proposed architecture. The overhead associated with the control circuitry is relatively small and simple. Reduction in power is obtained because of two counts, one is the reduction in the number of adders used thereby reducing the number of operations and the other is the reduction in switching activity of the adders based on the input data dependency. The adder array is realized by taking into consideration of the redundancies available within the distributed coefficient matrix. This results in a reduced number of adders to realize the array. A total of 26 adders were used to realize the coefficients, whereas 35 adders were used in [3]. A reduction of 91.56% is achieved in terms of number of adders with respect to direct implementation and about a 25.7% reduction compared to [3]. A comparison among the number of adders needed for proposed architecture with respect to direct implementation and the architecture in [3] are shown in Table 1.

Table 1. Comparison of number of adders required to realize of the coefficients between the proposed architecture and other published ones.

Type of approach	Number of adders	Percentage saving in VLSI area w. r. t. direct implementation
Direct implementation	308	---
NEDA	35	88.64 %
Proposed approach	26	91.56 %

7. Conclusion

A low power architecture has been proposed for the implementation of 1-D Discrete Cosine Transform. The architecture considered the fact that neighboring pixel values of a vector in image or video data are close to each other in magnitude. Fixed coefficient matrix distribution concept has been used in the architecture. The concept of

even and odd grouping of DCT coefficient computations is used to reduce the number of adders needed. The new data dependent architecture has reduced the VLSI area by 91.56 % with respect to the direct implementation and about 25.7 % with respect to NEDA architecture. Switching activity of the dynamic adders is controlled based on the input data dependency. A considerable reduction in power due to reduced number of adders and reduced switching activity have been achieved.

8. References

[1] K. R. Rao and P. Yip, *Discrete Cosine Transform-Algorithms, Advantages, Applications*, Academic Press, Inc. London, 1990.

[2] T. Xanthopoulos, A. P. Chandrakasan, "A Low-Power DCT Core Using Adaptive Bitwidth and Arithmetic Activity Exploiting Signal Correlations and Quantization," *IEEE Journal of Solid-State Circuits*, pp. 740 -750, May 2000.

[3] A. M. Shams, W. Pan, A. Chandanandan, M. A. Bayoumi, "A Low Power High Performance Distributed DCT Architecture," *IEEE Computer Society Annual Symposium on VLSI*, p. 0026, April 2002.

[4] K. Kim and P. A. Beerel, "A High-Performance Low-Power Asynchronous Matrix-Vector Multiplier for Discrete Cosine Transform," *Proceedings of the First IEEE Asia Pacific Conference on ASICs*, pp. 135-138, Aug. 1999.

[5] W. H. Chen, C. H. Smith, and S. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Transactions on Communications*, vol. COM-25, pp.1004-1009, Sept. 1977.

[6] M. T. Sun, L. Wu, and M. L. Liou, "A Concurrent Architecture for VLSI Implementation of Discrete Cosine Transform," *IEEE Transactions on Circuits Systems*, vol. CAS-34, pp. 992- 994, Aug. 1987.

[7] M.T.Sun, T.C.Chen, and A.M.Gottlieb, "VLSI Implementation of a 16 X 16 Discrete Cosine Transform," *IEEE Transactions on Circuits Systems*, vol. CAS-36, no.4, pp.610-617, April 1989.

[8] 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 Journal of Solid-State Circuits*, vol.27, pp. 492-498, April 1992.

[9] H. Fujiwara, M. L. Liou, M. T. Sun, K. M. Yang, M. Maruyama, K. Shomura, K. Ohyama, "An all-ASIC Implementation of a Low Bit-Rate Video Codec," *IEEE Transactions on Circuits Systems: Video Technology*, vol.2, pp.123-133, June 1992.

[10] Y-H.Chan and W-C.Siu, "On the Realization of Discrete Cosine Transform Using the Distributed Arithmetic," *IEEE trans. on Circuits and Systems-1: Fundamental Theory and Applications*, vol.39, no.9, pp.705-711, Sept.1992.

[11] A. P. Chandrakasan, and R. W. Brodersen, Minimizing Power Consumption in Digital CMOS Circuits, *Proceedings of IEEE*, vol. 83, no. 4, pp. 498-523, April 1995.

[12] K. Muhammad and K. Roy, "Switching Characteristics of Generalized Array Multiplier Architectures and Their Applications to Low Power Design," *Proc. Int. Conf. Comput. Design (ICCD'99)*, pp.230-235, Oct. 10-13, 1999.

[13] S. Yu, E. E. Swartzlander, "DCT Implementation with Distributed Arithmetic," *IEEE Transactions on Computers*, v 50, n 9, Sept., 2001, p 985-991.

[14] Y. Hong, Z. Hou, "Fast Algorithm of the DCT and IDCT for VLSI Implementation," *Proceedings of Int. Conf. Signal Processing - ICSP*, vol. 1, pp. 637-640, 1996.

[15] S. Perri, P. Corsonello, G. Cocorullo, "VLSI Circuits for Low-Power High-Speed Asynchronous Addition," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 5, pp. 608-613, 2002.

[16] S.Masupe and T.Arsalan, "Low Power Order Based DCT Processing Algorithm," *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 2, pp. II5-II8, 2001.

[17] Kyung- Wook Shin, Heung Woo Jean, Yeong Seum Kang, "Efficient VLSI Implementation of Vector-Radix 2-D DCT Using Mesh-Connected 2-D Array," *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 47-50, 1994.