International Journal of Electrical Electronics Computers & Mechanical Engineering (IJEECM)
ISSN: 2278-2808
www.ijeecm.org Volume 4 Issue 12 ǁ Dec. 2016

IJEECM journal of Electronics and Communication Engineering (ijeecm-jec)

Delay Efficient ACT Module Implementation Using VLSI Arithmetic Circuit

Pidugu Gopi[1], SVNK Pavan Kumar P[2]

[1,2] *Electronics and Communication Engineering Department,*

[1,2] Pydah Kaushik College Of Engineering, Visakhapatnam, A.P., India

Abstract:
The discrete cosine transform (DCT) is a widely-used and important signal processing tool employed in a plethora of applications. Typical fast algorithms for nearly-exact computation of DCT require floating point arithmetic, are multiplier intensive, and accumulate round-off errors. Recently proposed fast algorithm arithmetic cosine transform (ACT) calculates the DCT exactly using only additions and integer constant multiplications, with very low area complexity, for null mean input sequences. The ACT can also be computed non-exactly for any input sequence, with low area complexity and low power consumption, utilizing the novel architecture described. However, as a trade-off, the ACT algorithm requires 10 non-uniformly sampled data points to calculate the eight-point DCT. This requirement can easily be satisfied for applications dealing with spatial signals such as image sensors and biomedical sensor arrays, by placing sensor elements in a non-uniform grid. In this work, a hardware architecture for the computation of the null mean ACT is proposed, followed by a novel architecture that extend the ACT for non-null mean signals. All circuits are implemented and tested and synthesized using the Xilinx ISE14.7.

KEYWORDS: DCT, ACT, EIGHT-POINT DCT, AFT,

I. INTRODUCTION

THE discrete cosine transform (DCT) was first proposed by Ahmed et al. in 1974 and published in IEEE Transactions on Computers [1]. It has since attracted much attention in the computer engineering community [2], [3], [4], [5]. In particular, the eight-point DCT and its variants, in the form of fast algorithms, has been widely adopted in several image and video coding standards [6] such as JPEG, MPEG 1/2, and H.261-5 [7]. Some applications which use image and video compression include automatic surveillance [8], geospatial remote sensing [9], traffic cameras [10], homeland security [11], satellite based imaging [12], unmanned aerial vehicles [13], automotive [14], multimedia wireless sensor networks [15], the solution of partial differential equations [16] etc. A particular class of fast algorithms is constituted by the arithmetic ransforms. An arithmetic transform is an algorithm for low-complexity computation of a given trigonometric transform, based on number-theoretical results. A prominent example is the arithmetic Fourier transform (AFT) proposed by Reed et al. [17], [18]. The AFT allows multiplication-free calculation of Fourier coefficients using number-theoretic methods and non-uniformly sampled inputs. A feature of the AFT is its uitability for parallel implementation [17], [18]. Recently, an arithmetic transform method for the computation of the DCT, called the arithmetic cosine transform (ACT) was proposed in [19]. The ACT can provide a multiplication-free framework and leads to the exact computation of the DCT—provided that the input signal has null-mean and is non-uniformly sampled [19]. The computational gains of the ACT are only possible when its prescribed nonuniformly sampled data is available. Classically the required non-uniform samples are derived by means of interpolation over uniformly sampled data [19]. Such interpolation implies a computational overhead. Another aspect of the ACT is that, for arbitrary input signal, it requires the computation of the input signal mean value [19]. Usually, the mean value is computed from uniformly sampled data [19]. In fact, this dependence on uniformly sampled data has been precluding the implementation of the ACT based exclusively on non-uniformly sampled data. On the other hand, the requirement for non-uniform samples can be satisfied when spatial input signals are considered. In spatial signal processing, non-uniformly sampled signals can be directly obtained, without interpolation using a non-uniform placement of sensors [7], [20]. This motivates the search for architectures which could solely rely on non-uniformly sampled inputs. In this paper we address two main problems: (i) the proposition of a method to obtain the mean value of a given input signal from its non-uniform samples as prescribed by the ACT and (ii) the introduction of efficient architectures for calculation of the eight-point DCT based on the ACT, operating on non-uniformly sampled data only. This leads to designs with

low computational complexity. Having ACT architectures that compute 1-D DCT can be utilized as a building block to implement such 2-D DCT architectures that take inputs from sensors placed on a non uniform grid. Two architectures based on the ACT are sought, being referred to as Architectures I and II. Architecture I provides the hardware implementation of the ACT algorithm proposed in [19], and calculates the DCT with exact precision for null mean eight-point sequences. The proposed Architecture I is designed to require only additions and multiplications by integers. Thus, no source of intrinsic computation error is present, such as rounding-off and truncation. Therefore, area consuming hardware multipliers are not necessary. We propose Architecture II that implements the novel modified ACT algorithm for DCT calculation of arbitrary, non-null-mean input signals, using 11 hardware multiplications. Both architectures require only non-uniformly sampled inputs.

II.THE ARITHMETIC COSINE TRANSFORM

The usual input sequence to the DCT can be considered as uniform samples of a continuous input signal vðtÞ. This results in an N-point column vector $\mathbf{v} = \{v_n\}_{n=0}^{N-1}$ which has its DCT denoted by the N-point column vector $\mathbf{V} = \{V_k\}_{k=0}^{N-1}$. To calculate V, the ACT algorithm requiresnon-uniformly sampled points of the continuous input signal *v(t)* Þ [19]. These points are given by $r = \frac{2mN}{k} - \frac{1}{2}$ where k ¼ 1; 2; . . .;N _ 1, and m ¼ 0; 1; . . . ; k _ 1 [19]. We can define the set R as

R={ Set of all values of r}

It is important to notice that the values of r are not necessarily integer. In fact, they are expected to be fractional. If the signal of interest has zero mean, then the ACT algorithm can be used to calculate the DCT coefficients as follows. First, let the ACT averages Sk, k ¼ 1; 2; . . .;N _ 1, of the non-uniform sampled inputs be defined as [19]:

$$S_k \triangleq \frac{1}{k}\sum_{m=0}^{k-1} v_{2m\frac{N}{k}\frac{1}{2}}, \quad k = 1, 2, \ldots N-1.$$

The ACT averages can be employed to computed DCT coefficients according to [19]:

$$V_k = \sqrt{\frac{N}{2}} \sum_{l=1}^{\lfloor \frac{N-1}{k} \rfloor} \mu(l) \cdot S_{kl}, \quad k = 1, 2, \ldots, N-1,$$

where mð_Þ is theM€obius function . The derivation of the ACT [19] utilizes the M€obius inversion formula. Because the M€obius function values are limited to f_1; 0;þ1g, (3) results in no additional multiplicative complexity. In actice, input sequences are not always null mean, therefore a correction term is necessary to (3). In [19] an expression suitable for the non-null mean signals is given as:

$$V_k = \sqrt{\frac{N}{2}} \sum_{l=1}^{\lfloor \frac{N-1}{k} \rfloor} \mu(l) \cdot S_{kl} - \sqrt{\frac{N}{2}} \bar{v} \cdot \mathrm{M}\left(\left\lfloor \frac{N-1}{k} \right\rfloor \right),$$

where $\mathrm{M}(n) \triangleq \sum_{m=1}^{n} \mu(m)$ is the Mertens function [19] and _v is the mean value of the uniformly sampled input sequence.

III. EXISTING ALOGORITHM

3.1 Mean Value Calculation

Although (4) leads to the DCT coefficients of non-null mean input signals, it requires the knowledge of quantity _v, which could be calculated straightforwardly from the N uniform samples in v. Since uniform samples are not available, _v should be directly calculated from non-uniform samples. The non-uniform samples are related to the uniform samples according to the interpolation scheme given by [19]:

$$v_r = \sum_{n=0}^{N-1} w_n(r) \cdot v_n, \quad r \in R,$$

where $w_n(r)$ is the interpolation weight function expressed by:

$$w_n(r) = \frac{1}{2N}\left[\mathrm{D}_{N-1}\left(\frac{\pi}{N}(n+r+1)\right) + \mathrm{D}_{N-1}\left(\frac{\pi}{N}(n-r)\right)\right], \quad n = 0, 1, \ldots, N-1,$$

And

$$\mathrm{D}_N(x) = \frac{\sin\left((N+1/2)x\right)}{\sin\left(x/2\right)}$$

denotes the Dirichlet kernel [21, p. 312]. Here, the set R is defined in (1). More compactly, (5) can be put in matrix form. Indeed, we can write vr ¼ W_ v, where vr ¼ ½vr_r2R is a column vector containing the required non-uniform samples, $\mathbf{W} = \{w_n(r)\},\ n = 0, 1, \ldots, N-1,\ r \in R,$ is the implied interpolation matrix. For the particular case of the eight-point ACT, the following 10 non-uniform sampling instants are required.

$$r \in R = \left\{ -\frac{1}{2}, \frac{25}{14}, \frac{13}{6}, \frac{27}{10}, \frac{7}{2}, \frac{57}{14}, \frac{29}{6}, \frac{59}{10}, \frac{89}{14}, \frac{15}{2} \right\}.$$

Moreover, matrix W is found to possess full column rank. Thus, its Moore-Penrose pseudo-inverse Wþ is the left inverse ofW[22, p. 93]. Therefore, we obtain

$$\mathbf{v} = \mathbf{W}^{+} \cdot \mathbf{v_r}.$$

Consequently, the mean value of v can be determined exclusively rom the non-uniform samples, according to

$$\bar{v} = \frac{1}{8}\mathbf{w} \cdot \mathbf{v_r},$$

where w is the eight-point vector of the sums of each column of Wþ. Scaled vectorw=8 has constant elements given by

$$\frac{1}{8}\mathbf{w}^{\top} = \begin{bmatrix} 0.131763492716950 \\ 0.498388117552161 \\ -0.313306526814540 \\ 0.018837637958148 \\ 0.389746948996966 \\ -0.178465262210960 \\ 0.166302458810496 \\ 0.269801852271683 \\ -0.131541981375149 \\ 0.148473262094246 \end{bmatrix},$$

were the superscript > denotes the transposition operation

Matrix Factorization of ACT

In view of (2) and (7), (4) can be interpreted as the sought relation between V and vr. Thus, we can consider a Transformation matrix T relating these two vectors. Notice that T is not a square matrix. Since k ¼ 1; 2; . . .;N _ 1, the size of T is ðN _ 1Þ _ jRj, where jRj is the number of elements of R. This transformation matrix returns all the DCT components, except the zeroth one, according to

$$[\,V_1 \quad V_2 \quad \cdots \quad V_{N-1}\,]^{\top} = \mathbf{T} \cdot \mathbf{v_r}.$$

Notice that. $V_0 = \sqrt{N} \cdot \bar{v}$.

For N ¼ 8, matrix T has size 7_10 and admits the following matrix factorization:

$$\mathbf{T} = 2 \cdot \mathbf{Mo} \cdot \mathbf{D_1} \cdot \mathbf{S} + \mathbf{Me} \cdot \mathbf{W}^{+},$$

$$\mathbf{Mo} = \begin{bmatrix} 1 & -1 & -1 & 0 & -1 & 1 & -1 \\ 0 & 1 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{D_1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{7} \end{bmatrix},$$

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 1 \\ 1 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 \end{bmatrix},$$

where 17 is the 7_7 matrix of ones. In (8), matrix D1 and S are related to (2). Matrix Mo contains the values of the M€obius functions as required in (3). The second term in the right-hand side of (8) accounts for the Mertens functions and the mean value calculation as required in (4) and (7).

Existing architectures

In this section, above discussed methods are employed to furnish two novel low complexity architectures, which take only non-uniformly sampled inputs. Integer multiplications, which are exact in nature, are realized using simple shift- dd structures. The designs are fully pipelined by judicious sertion of registers at internal nodes, leading to low critical path delay at the cost of latency.

Architecture I

The ACT expressions for null mean signals in (2) and (3) can be implemented for N ¼ 8 as shown in . We refer to this design as Architecture I. The eight-point null mean ACT block admits the 10 non-uniformly sampled inputs corresponding to (6). Constant multiplications by two are implemented as left-shift operations; and the fractional constant multipliers 1=2; 1=3; 1=4; . . . are converted to integers by scaling them by the least common multiple of their denominators: 420. The integer constant multipliers can be implemented as Booth encoded shift-and-add structures making the architecture multiplier free, and the outputs of the block are scaled by 420 . $\sqrt{2/N} = 210$This architecture is useful in applications that have null mean input sequences, and can be implemented with very low computational and area complexity.

Architecture II

The proposed method in Section 3 for the computation of _v from the non-uniformly sampled 10-point signal can be implemented as shown in Fig. 1b. We will refer to it as the mean calculation block, which computes (7). The correction term associated to the Mertens function required in (4) isshown in Fig. 2a. A combination of (i) this particular block, (ii) the Architecture I block, and (iii) the mean calculation block yields the proposed Architecture. Note that calculation of the DCT coefficients using the null mean ACT block can also be achieved by subtracting the mean _v from its inputs. However, Architecture II has a lower computational complexity when compared to such alternative. Computation complexity of both Architecture I and Architecture II are listed in Table 1 in terms of constant multipliers and two-input adders. Integer constant multiplications are implemented as shift-and-add structures, therefore are not counted as multipliers. Note that the adder count also include the adders required for the Booth encoded structures

IV.PROPOSED ALGORITHM

Ladner-Fischer adder:

Ladner- Fischer parallel prefix adder was developed by R. Ladner and M. Fischer in 1980. Ladner-Fischer prefix tree is a structure that sits between Brent-Kung and Sklansky prefix tree. The LF adder has minimum logic depth but it has large fan-out. Ladner- Fischer adder has carry operator nodes. The delay for the type of Ladner-Fischer prefix tree is *log2n+1*. Fig. shows the 16-bit LF adder.

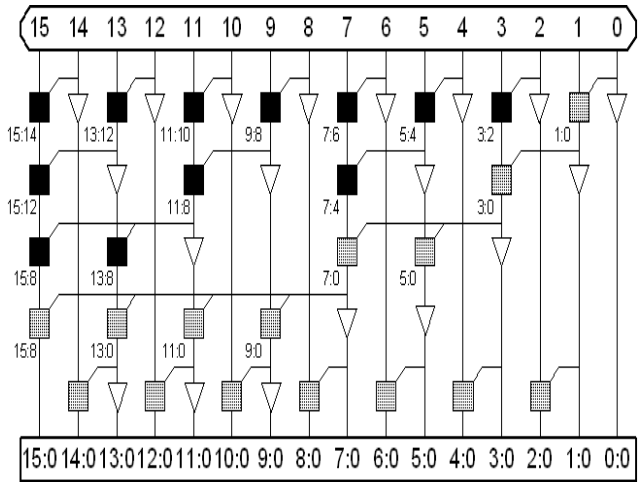


Fig 1 .16-bit Ladner- Fischer Adder:

Carry Stages: log2 n;
The number of cells: (n/2).log2n;
Maximum fan-out: n/2.

***Parallel-Prefix adders*:**

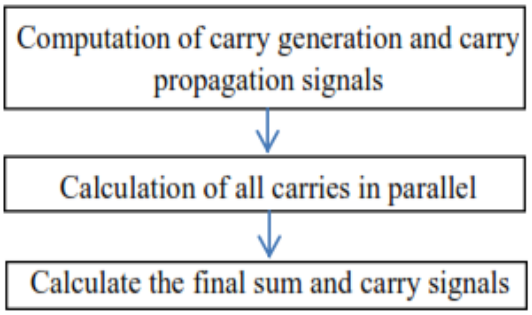
Parallel-Prefix adders perform parallel addition i.e. most important in microprocessors, DSPs, mobile devices and other high speed applications. Parallel-Prefix adder reduces logic complexity and delay thereby enhancing performance with factors like area and power. Therefore the Parallel-Prefix adders are requisite element in the high speed arithmetic circuits and popular since twenty years. Parallel prefix computation carries out three necessary or vital steps:

1) Computation of carry generation & carry propagation signals by using no. of input bits.

2) Calculating all the carry signals in parallel that is called prefix computation.

3) Evaluating total sum of given inputs.

These steps are given in fig. that is given above.



The process or three steps that is carried out in the parallel prefix addition is as follows:

1. Computation of carry generation, propagation signals:

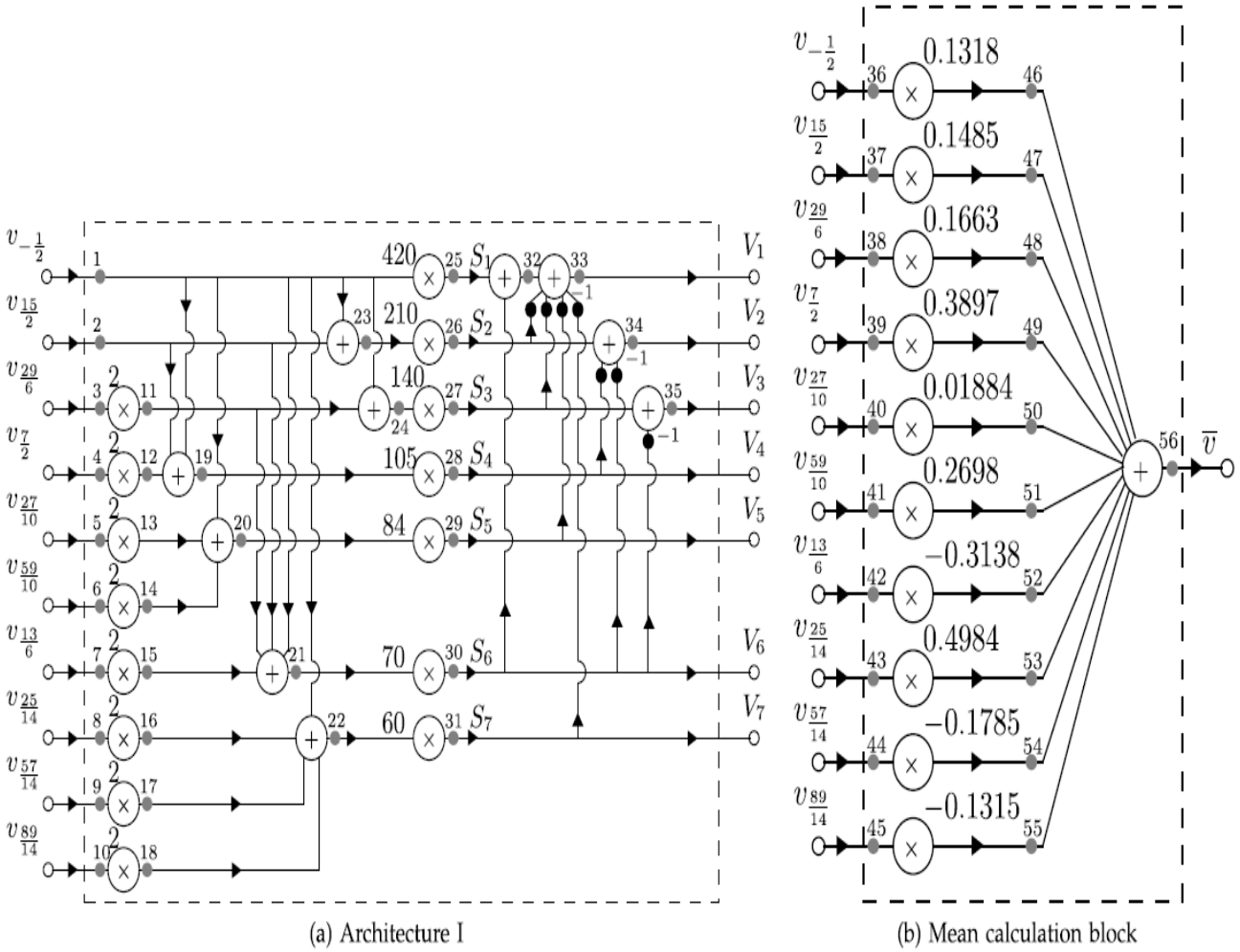


Fig. 2. (a) Null mean ACT and (b) mean calculation block

Previous carry is calculated to the next bit is called propagate signal and generate is to generate the carry bit below are the signals:

$$G_i = A_i \cdot B_i \quad \text{.....(1)}$$

$$P_i = A_i \oplus B_i \quad \text{..... (2)}$$

2. Calculation of all carry signals:

$$G_{i:j} = G_{i:k} + P_{i:k} \cdot G_{k-1:j} \quad \text{.....(3)}$$

$$P_{i:j} = P_{i:k} \cdot P_{k-1:j} \quad \text{..... (4)}$$

3. Calculation of Final Sum:

$$S_i = P_i \oplus G_{i-1:0} \quad \text{..... (5)}$$

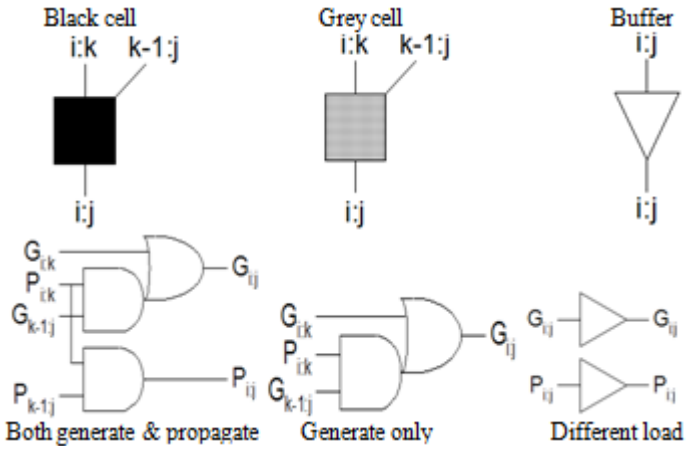


Fig 3: Black, Grey cells, Buffer and there Schematics

V. IMPLEMETATION AND RESULTS

We implemented Ladner- Fischer parallel prefix adder architectures described in the previous section. Ladner-Fischer prefix tree is a structure that sits between Brent-Kung and Sklansky prefix tree. The LF adder has minimum logic depth but it has large fan-out. Ladner- Fischer adder has carry operator nodes. The delay for the type of Ladner-Fischer prefix tree is *log2n+1*. Fig. shows the 16-bit LF adder.

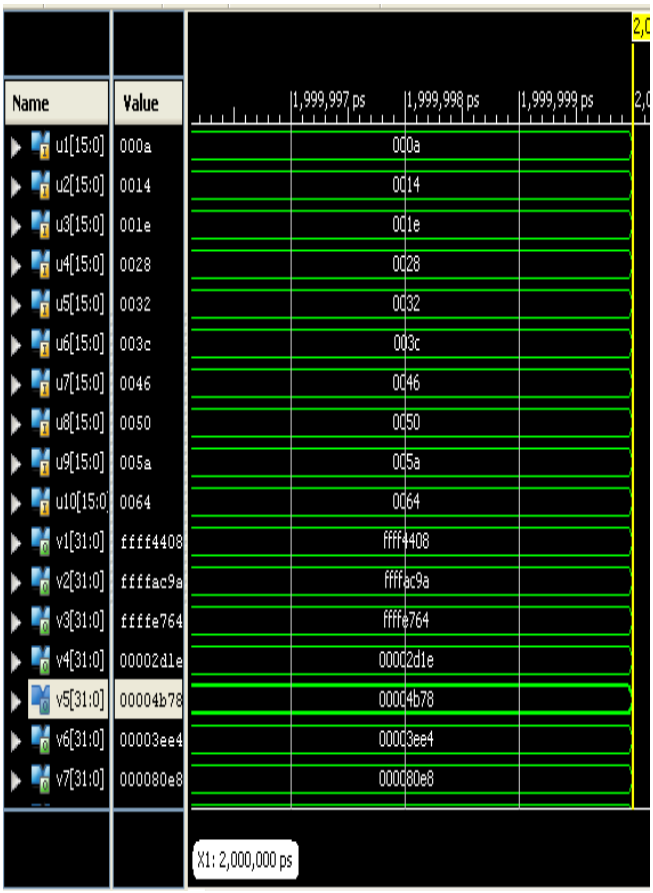


Fig 4. Simulation results of ACT

Parallel-Prefix adders perform parallel addition i.e. most important in microprocessors, DSPs, mobile devices and other high speed applications. Parallel-Prefix adder reduces logic complexity and delay thereby enhancing performance with factors like area and power. Therefore the Parallel-Prefix adders are requisite element in the high speed arithmetic circuits

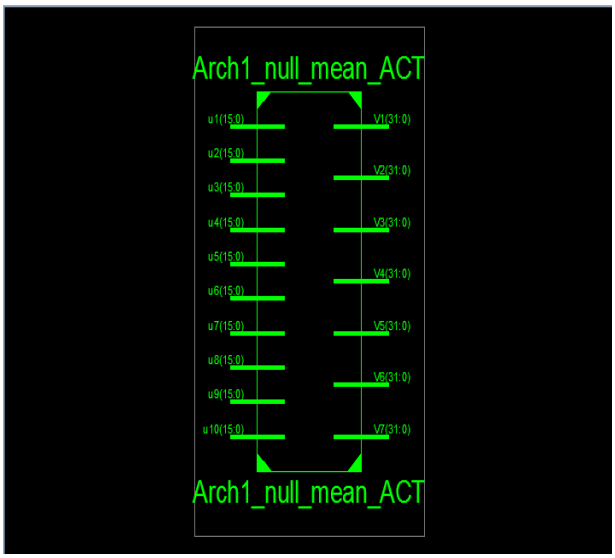


Fig 5 RTL Schematic of ACT

VI. CONCLUSION

The ACT algorithm is suitable for calculating the eight point DCT coefficients exactly using only adders and integer constant multiplications, also with low computational complexity. ACT architectures for null mean inputs as well as for non-null mean inputs are proposed, implemented and tested on XilinxISE14.7. The average percentage error and PSNR were adopted as figures of merit to assess the measured results. Results show that even for lower fixed point word-lengths, the implementations lead to acceptable margins of error. The resource utilization for various fixed point implementations indicates a trade-off between accuracy and device resources (chip area, speed, and power). It is the first step towards new research on low power and low complexity computation of the DCT by means of the recently proposed ACT.

REFERENCES

[1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," IEEE Trans. Comput., vol. 23, no. 1, pp. 90–93, Jan. 1974.

[2] C. Chakrabarti and J. J_aJ_a, "Systolic architectures for the computation of the discrete Hartley and the discrete cosine transforms based on prime factor decomposition," IEEE Trans. Comput., vol. 39, no. 11, pp. 1359–1368, Nov. 1990.

[3] F. A. Kamangar and K. R. Rao, "Fast algorithms for the 2-D discrete cosine transform," IEEE Trans. Comput., vol. 31, no. 9, pp. 899–906, Sep. 1982.

[4] H. Kitajima, "A symmetric cosine transform," IEEE Trans. Comput., vol. 21, no. 4, pp. 317–323, Apr. 1980.

[5] S. Yu and E. Swartziander Jr, ,"DCT implementation with distributed arithmetic," IEEE Trans. Comput., vol. 50, no. 9, pp. 985–991, Sep. 2001.
[6] V. Britanak, P. Yip, and K. R. Rao, Discrete Cosine and Sine Transforms. Amsterdam, The Netherlands: Academic Press, 2007.
[7] N. Romaand L. Sousa,"Efficient hybrid DCT-domain algorithm for video spatial downscaling," EURASIP J. Adv. Signal Process., vol. 2007, no. 2, pp. 30–30, 2007.
[8] H. Lin and W. Chang, "High dynamic range imaging for stereoscopic scene representation," in Proc. 16th IEEE Int. Conf. Image Process., Nov. 2009, pp. 4305–4308.
[9] E. Magli and D. Taubman, "Image compression practices and standards for geospatial information systems," in Proc. IEEE Int. Geosci. Remote Sens. Symp., Jul. 2003, vol. 1, pp. 654–656.
[10] M. Bramberger, J. Brunner, B. Rinner, and H. Schwabach, "Realtime video analysis on an embedded smart camera for traffic surveillance," in Proc. 10th IEEE Real-Time Embedded Technol. Appl. Symp., May. 2004, pp. 174–181.
[11] C. F. Chiasserini and E. Magli, "Energy consumption and image quality in wireless video-surveillance networks," in Proc. 13th IEEE Int. Symp. Pers., Indoor Mobile Radio Commun., Sep. 2002, vol. 5, pp. 2357–2361.
[12] T. Tada, K. Cho, H. Shimoda, T. Sakata, and S. Sobue, "An evaluation of JPEG compression for on-line satellite images transmission," in Proc. Int. Geosci. Remote Sens. Symp., Aug. 1993, pp. 1515–1518.
[13] B. Bennett, C. Dee, and C. Meyer, "Emerging methodologies in encoding airborne sensor video and metadata," in Proc. IEEE Military Commun. Conf., Oct. 2009, pp. 1–6.
[14] S. Marsi, G. Impoco, A. Ukovich, S. Carrato, and G. Ramponi, "Video enhancement and dynamic range control of HDR sequences for automotive applications," EURASIP J. Adv. Signal Process., vol. 2007, p. 080971, 2007.
[15] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury,"A survey on wireless multimedia sensor networks," Comput. Netw., vol. 51, no. 4, pp. 921–960, 2007.
[16] J. G. Proakis and D. G. Manolakis, Digital Signal Processing. Upper Saddle River, NJ, USA: Prentice-Hall, 2007.
[17] I. S. Reed, D. W. Tufts, X. Yu, T. K. Truong, M. T. Shih, and X. Yin, "Fourier analysis and signal processing by use of the M€obius inversion formula," IEEE Trans. Acoust., Speech Signal Process., vol. ASSP-38, no. 3, pp. 458–470, Mar. 1990.
[18] I. S. Reed, M. T. Shih, T. K. Truong, E. Hendon, and D. W. Tufts, "A VLSI architecture for simplified arithmetic Fourier transform algorithm," IEEE Trans. Signal Process., vol. 40, no. 5, pp. 1122–1133, May 1992.
[19] R. J. Cintra and V. S. Dimitrov, "The arithmetic cosine transform: Exact and approximate algorithms," IEEE Trans. Signal Process., vol. 58, no. 6, pp. 3076–3085, Jun. 2010.
[20] E. J. Tan, Z. Ignjatovic, and M. F. Bocko, "A CMOS image sensor with focal plane discrete cosine transform computation," in Proc. IEEE Int. Symp. Circuits Syst., May. 2007, pp. 2395–2398.
[21] S. G. Krantz, Real Analysis and Foundations. Boca Raton, FL, USA: Chapman & Hall, 2005.
[22] I. C. F. Ipsen, Numerical Matrix Analysis: Linear Systems and Least Squares, Philadelphia, PA, USA: SIAM, 2009.
[23] J. Liang and T. D. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," IEEE Trans. Signal Process., vol. 49, no. 12, pp. 3032–3044, Dec. 2001.
[24] T. D. Tran, "The binDCT: Fast multiplierless approximation of the DCT," IEEE Signal Process. Lett., vol. 7, no. 6, pp. 141–144, Jun. 2000.
[25] D. Gong, Y. He, and Z. Cao, "New cost-effective VLSI implementation of a 2-D discrete cosine transform and its inverse," IEEE Trans. Circuits Syst. Video Technol., vol. 14, no. 4, pp. 405–415, Apr. 2004.
[26] A. Shams and M. Bayoumi, "A 108 Gbps, 1.5 GHz 1D-DCT architecture," in Proc. IEEE Int. Conf. Appl.-Specific Syst., Archit., Process., 2000, pp. 163–172.
[27] S. Ghosh, S. Venigalla, and M. Bayoumi, "Design and implementation of a 2D-DCT architecture using coefficient distributed arithmetic [implementaion read implementation]," in Proc. IEEE Comput. Soc. Annu. Symp. VLSI, May. 2005, pp. 162–166.
[28] V. S. Livramento, B. G. Moraes, B. A. Machado, and J. L. Guntzel, "An energy-efficient 8_8 2-D DCT VLSI architecture for batterypowered portable devices," in Proc. IEEE Int. Symp. Circuits Syst., 2011, pp. 587–590.