International Journal of
Scientific Engineering and Technology Research

www.ijsetr.com

ISSN 2319-8885
Vol.06,Issue.27
August-2017,
Pages:5219-5223

An Efficient Usage of D-Latch Approach for Carry Select Adder in DCT for Image Processing Applications

VENKATESWARARAO NARALASETTY[1], N. SURESHBABU[2]
[1]PG Scholar, Chirala Engineering College, Chirala, AP, India.
[2]Associate Professor, Chirala Engineering College, Chirala, AP, India.

Abstract: Discrete cosine transforms (DCT) are used for reducing the difficulty of the circuit. The existing algorithm is useful for the estimate of DCT of small length transforms and some of the algorithms are non-orthogonal. The addition process is performed at the input stage that is useful for finding out the length of estimated DCT and in this paper we use algorithms that are used to obtain orthogonal estimate of DCT. The proposed algorithms are obtain by the symmetries of basic vectors of DCT and are used in scaling of hardware and software implementations of DCT at higher lengths. The existing 8point DCT algorithms are obtained by using the proposed algorithms and it make use of DCT of power two lengths. By using matrix decomposition the algorithm for the existing DCT will be defined. The proposed DCT is having better image and video compression. The proposed algorithm has low complexity while calculation when compared to other algorithms. In this paper for parallel calculation it can calculate 32 point DCT or 16 point DCT by placing them in parallel or 8 point DCT by using marginal control overhead. So the calculation of DCT is taken parallel which reduces the complexity of the circuit. The approximate DCT which is proposed having less circuit complexity i.e., the hardware requirement is less and maintenance is easy as compared with existing DCT. The below results proves that FPGA implementation is better than existing one. The 64, 128 and 256 bit DCT are proposed in this paper.

Keywords: DCT, Xilinx, Verilog.

I. INTRODUCTION

DCT's are the one used mostly in compressing data which is in image or video format. The 8point DCT will have less complexity than the 16 point DCT. Multiplication is the method which is more power and time hugry block whose direct calculation is eliminated in this method. Signed DCT is one of such method proposed by "Hawel". Bouguezel-Ahmad-Swamy continued the further processing. These people propose a well DCT by giving the vectors as 0, ½, and 1. Zero, one are the transforms elements that are proposed by Cintra and Bayer. Finally kernel and shave introduces their methods and proved that it performs enhanced than those methods that are stated above. As the amount of points goes on increasing in DCT the circuit difficulty also increases, so the requirement for resembling the larger point DCT is necessary. In the case of the application video calling it requires more amount of DCT blocks upto 32*32 for achieving the compression ratio more. The size of DCT will goes on increasing in some cases of image processing such as tracking and encryption.

Due to the difficulty of increase in size of DCT a method called "Cintra" that introduces integer transforms and is apply to several block lengths of DCT. Potluri et al. proposed an approximation for 8 point DCT and it computes the result with fewer amount of additions. Another method for 8 point approximation DCT has been proposed by Cintra et al with low complexity of hardware. Appropriate extension of integer DCT for 128 and 256 point proceed with multiplication approximation and it is proposed by Bouguezel. Walsh- hadamard transform is one of the methods proposed for implementing DCT of upper order. While going for estimate of DCT we should remain three points. The primary one is the approximate DCT should have less hardware requirements and the other one is while compressing it should have a smaller amount of errors and the data will not loss completely and preferably should be orthogonal. It should work for modern video calling standards and other applications like tracking, surveillance and encryption. But the above necessities are not fulfilled by the existing DCT like their size and orthogonality. The main reason for maintain orthogonal DCT is to find the inverse and forward structures of DCT by using inverse matrix and kernel matrix. The inverse matrix will obtained by using the orthogonal transform of DCT.

II. DISCRETE COSINE TRANSFORMS

Used in many standards like compression of image or video this is in JPEG, removing and retrieving the data back when it is necessary, storage purpose. Some of the applications like video conferencing, wireless video DCT is used Compression. Here we use are of two types. It uses the bits only that are essential for representing a signal and lossless compression is the another one that uses ratio around 2-3:1 to recover data. 1000:1 is the compression ratio for video and wireless video. The degradation of a signal is done by compression. Statistical, temporal and spatial are the methods used by compression. Statistical method uses arithmetic, Lempel-Ziv. Spatial method uses quantization of vectors, wavelets, sub bands. Temporal method compresses a moving data. Dividing the frame into luminance (measures intensity), chrominance (color) images (YUV). The intensity of the picture is done through 16*16 macro blocks or four 8*8 macro blocks. Color of a picture is done through two 8*8 blocks. Lossless code is obtained by quantized co-efficients. Transform coefficients are quantized. An 8*8 DCT requires 16 DCT that calls 16 functions that direct to inappropriate 8-point DCTs.1-D DCT explains the extension as 2N-point DCT having algorithm #1 for 1-D DCT. Set calculate using 2N-point DFT Set. This requires N+N log2 (2N) complex multiplies. Another method that computes with less multiplies by using a shorter DFT. Consider the 1-D DFT and the sum part is halved into even samples and the odd samples are separated_ note that_ G[k} and H[k] are N-point DFT's. Set Therefore, Algorithm #2 Set_ calculate the N-point DFT of The condition in front of Greal[k].

III. DCT APPROXIMATION

The transforms (DCT) matrix elements is given by:

$$c(i,j) = \epsilon_i \sqrt{\frac{2}{N}} \cos \frac{(2j+1)i\pi}{2N} \tag{1}$$

where $0 \leq i,j \leq N-1$, $\epsilon_0 = 1/\sqrt{2}$ and $\epsilon_i = 1$ for i>0. Exact DCT distinguish it from approximated forms of DCT. For $k \in [0,(N/2)-1]$ and $i = 2k$ we can

$$c(2k,j) = \epsilon_{2k} \sqrt{\frac{2}{N}} \cos \frac{(2j+1)2k\pi}{2N} \tag{2}$$

Since, $\epsilon_{2k} = \epsilon_k$, (2) can be rewritten as:

$$c(2k,j) = \epsilon_k \sqrt{\frac{2}{N}} \cos{(2j+1)k\pi N} \tag{3}$$

side of (3) corresponds to/-point DCT and its elements can be assumed to be $\sqrt{2}c(k,j)$, for $0 \leq j \leq (N/2)-1$. Therefore, the first / the correspondence size of DCT matrix having elements in even rows $N/2$ -point DCT matrix. The decomposition of $\boldsymbol{C}_N$can be performed as detailed. Using the even/odd symmetries of its row vectors, DCT matrix can be represented by the following matrix product.

$$\boldsymbol{C}_N = \frac{1}{\sqrt{2}} \boldsymbol{M}_N^{per} \boldsymbol{T}_N \boldsymbol{M}_N^{add} \tag{4}$$

Where $\boldsymbol{T}_N$ is a block sparse matrix expressed by:

$$\boldsymbol{T}_N = \begin{bmatrix} \boldsymbol{C}_{\frac{N}{2}} & \boldsymbol{0}_{\frac{N}{2}} \\ \boldsymbol{0}_{\frac{N}{2}} & \boldsymbol{S}_{\frac{N}{2}} \end{bmatrix} \tag{5}$$

Where $\boldsymbol{0}_{N/2}$ is the $((N/2)\times(N/2))$zero matrix. Block sub matrix / consists of odd rows of the $N/2$ first columns of $\sqrt{(2)}C_N . M_N^{per}$. is a permutation matrix expressed by:

$$\boldsymbol{M}_N^{per} = \begin{bmatrix} \boldsymbol{P}_{N-1,\frac{N}{2}} & \boldsymbol{0}_{1,\frac{N}{2}} \\ \boldsymbol{0}_{1,\frac{N}{2}} & \boldsymbol{P}_{N-1,\frac{N}{2}} \end{bmatrix} \tag{6}$$

Where $\boldsymbol{0}_{1,N/2}$is a row of zeros $N/2$ and $P_{N-1,N/2}$ is a matrix defined as:

$$\boldsymbol{P}_{N-1,\frac{N}{2}}^{(i)} = \begin{cases} \boldsymbol{0}_{1,\frac{N}{2}} & if \quad i = 1,3,5,\ldots,N-1 \\ \boldsymbol{I}_{\frac{N}{2}}\left(\frac{i}{2}\right) & if\; i = 0,2,4,\ldots,N-2 \end{cases} \tag{7}$$

Where $\boldsymbol{I}_{N/2}(i/2)$ is the $(i/2)$th row vector of the $((N/2)\times(N/2))$ identity matrix. Finally, the last matrix in (4), $\boldsymbol{M}_N^{add}$is defined by:

$$\boldsymbol{M}_N^{add} = \begin{bmatrix} \boldsymbol{I}_{\frac{N}{2}} & \boldsymbol{J}_{\frac{N}{2}} \\ \boldsymbol{I}_{\frac{N}{2}} & -\boldsymbol{J}_{\frac{N}{2}} \end{bmatrix}, \tag{8}$$

where $\boldsymbol{J}_{N/2}$ is an $((N/2)\times(N/2))$ zeros elsewhere.

Therefore, complexity of computation is reduced by using approximation of point DCT, $\boldsymbol{T}_N$. Let $\hat{C}_{N/2}$ and $\hat{s}_{N/2}$ denote the approximation matrices of $C_{N/2}$ and $\boldsymbol{S}_{N/2}$, respectively. The trade-off analysis shows the approximation $\boldsymbol{C}_8$ by where current state of 8-point DCT is done by using rounding-off operation methods. Observe (4) and (5), we note that $\boldsymbol{C}_8$ operates on sums while $\boldsymbol{S}_8$ operates on differences of the same pixel pairs. Therefore, if we replace $\hat{\boldsymbol{S}}_8$ by $\hat{\boldsymbol{C}}_8$, we shall have two advantages. The first is compression is done purely $\boldsymbol{S}_8$ and the next one is implementing is easy, and reconfigurable. $\hat{\boldsymbol{C}}_8$ We have investigated two other low-complexity alternatives, and in the following we discuss here three possible options of approximation $\boldsymbol{S}_8$: the first one is to approximate with the matrix having all zero elements.

This approximation values are different from the existed values of even-indexed and the odd part doesn't contain any information. Next one is approximating by an 8*8 matrix 1 will be present in each row and all other elements are zero. Here elements equal to 1 maximum of exact DCT in each row. The approximate transform is closer than the solution obtained by null matrix. Next one consists of approach $\boldsymbol{S}_8$by $\hat{\boldsymbol{C}}_8$ since $\boldsymbol{C}_8$ as well as $\boldsymbol{S}_8$ are sub matrices of/ and sum and differences of pixel pairs for an 8-point DCT. It has so many advanced properties like: regularity of the signal flow graph, orthogonality since $\hat{\boldsymbol{C}}_8$ is orthogonalizable. So there could be other possible for reduced complexities. But other solutions are not the potential for reconfigurablity what we achieve by replacement of by $\hat{\boldsymbol{S}}_8$. By observing these approximation approaches $\hat{\boldsymbol{C}}_N$ as$\boldsymbol{C}_N$:

$$\hat{\boldsymbol{C}}_N = \frac{1}{\sqrt{2}} \boldsymbol{M}_N^{per} \begin{bmatrix} \hat{\boldsymbol{C}}_{\frac{N}{2}} & \boldsymbol{0}_{\frac{N}{2}} \\ \boldsymbol{0}_{\frac{N}{2}} & \hat{\boldsymbol{C}}_{\frac{N}{2}} \end{bmatrix} \boldsymbol{M}_N^{add}. \tag{9}$$

As stated before, matrix / is orthogonalizable.

$$\boldsymbol{D}_N = \sqrt{\left(\hat{\boldsymbol{C}}_N \times \left(\hat{\boldsymbol{C}}_N\right)^t\right)^{-1}}, \tag{10}$$

where $(.)^t$ denotes matrix transposition. For data compression, we can use $C_N^{orth} = D_N \times \hat{C}_N$ instead of $\hat{\boldsymbol{C}}_N$ since $(\boldsymbol{C}_N^{orth})^{-1} = (\boldsymbol{C}_N^{orth})^t$. Since it is a matrix, can be done scaling in quantization process (without additional computational complexity).The process for the approximated DCT is stated in algorithm 1.

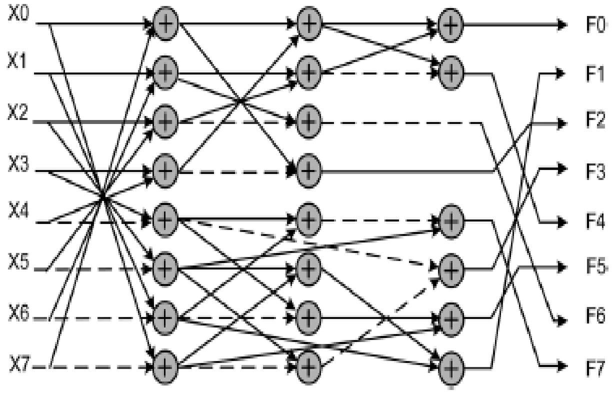


Fig1. signal flow graph (SFG) of $(\hat{C}_8)$.

The essential calculation unit of proposed DCT, $\hat{C}_8$ is given. The DCT computation is done and is shown in below diagram $\hat{C}_8$ is shown in Fig.1. For a given input sequence $\{\boldsymbol{X}(n)\}, n \in [0, N-1]$, coefficients of DCT are obtained by $\boldsymbol{F} = \hat{\boldsymbol{C}}_N \cdot \boldsymbol{X}^t$. An example of block diagram $\hat{\boldsymbol{C}}_{16}$is illustrated in fig. 2.

III. COMPLEXITY COMPARISON

To evaluate the computation for proposed/-point approximate DCT $N(\log_2 N - (1/4))$, we need to determine the cost of matrices quoted in (9). As shown in fig.1 the approximate 8-point DCT involves 22 additions. The involved in proposed DCT approximation and those competing approximations are shown in tableI. the requirement is lowest number of additions, and does not require any shift operations. Note that shift operation combinational components, and require only rewiring during hardware implementation. But it has indirect hardware complexity since shift-add operations lead to increase in bit-width which leads to higher hardware complexity of arithmetic units which follow the shift-add operations.

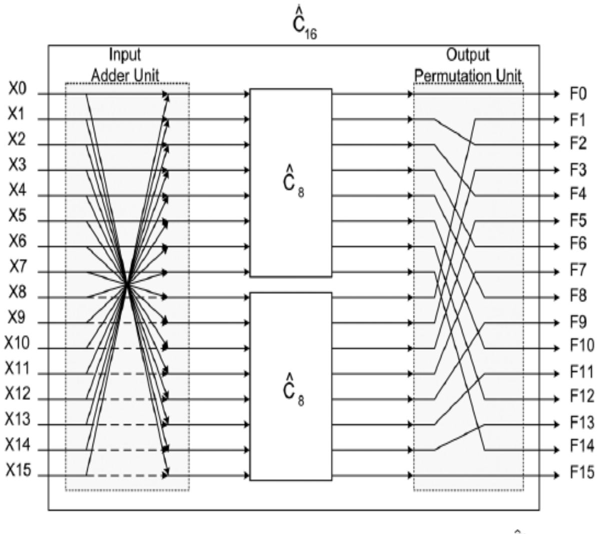


Fig2. Proposed block diagram for DCT $N = 16\ (\hat{\boldsymbol{C}}_{16})$.

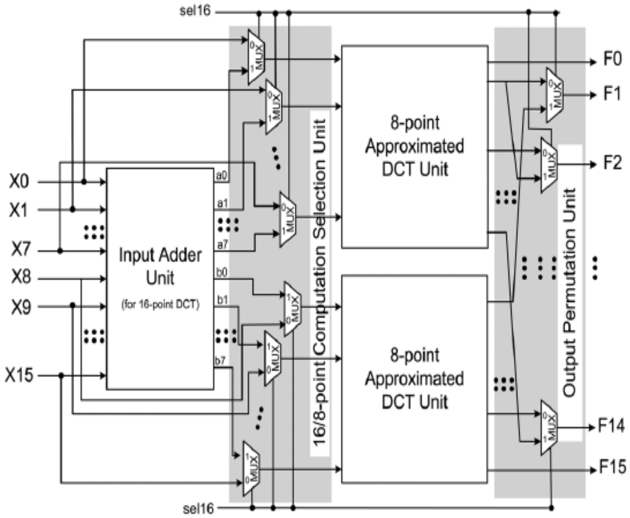


Fig3. Proposed reconfigurable architecture for approximate DCT.

A. Proposed Reconfiguration Scheme

Here it is proposed that a reconfigurable DCT structures which could be reused. The implementation of approximated 16-point DCT in fig.3. have of 3 blocks, in that there are two 8-point approximated DCT units which are 2 in number and a 16-point that generates $a(i)$and $b(i), i \in [1:7]$,. In the other sense, the output permutations unit uses 14 MUX to select and re-order the size of the output selected DCT. Specifically $Sel16 = 1$enables and $Sel16 = 0$ enables 8-point DCTs in parallel. Consequently the architecture if fig.3 allows the 16-point DCT or two 8-points in parallel.

B. CSLA Using D-Latch Logic

This method replaces the BEC circuit by D-latch. Latches are used to store 1-bit binary information. The latch is one of the sequential circuits so, their outputs are depends on the present inputs and previous inputs. Latch is also known as level sensitive, therefore when latch is enabled, and then the operation of latch will change according with input signal of the latch. The architecture of wished-for 16-bit Carry Select Adder is exposed in Fig. 3. In this we are using five different ripple carry adders with different bit size and D-Latch. In this proposed method uses only one adder in its place of using two separate adders in the regular carry select adder (CSLA) to reduce the area, and power consumption. With in 1 of the clk period both of the 2 additions are performed in the CSLA. A 2 bit LSB is used in the corresponding 16-bit adder. The first half which is nothing but the msp is 14 bit wide is works depends on the clock, the input carry performed addition when carry goes high. The carry input is assumed as zero while clock goes low and the sum of adder is stored in adder itself. From the Fig. it can understand that latch is used to store the sum and carry for Cin=1 and cin=0.Carry out from the previous stage i.e., for selecting final output carry and sum of 16-bit adder a multiplexer control signal is selected from least significant bit adder. If the actual carry input is one, then computed sum and carry latch is accessed and for carry input zero MSB adder is accessed is the output carry.

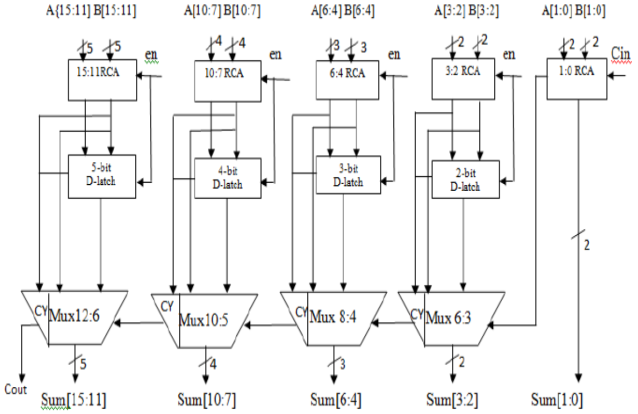


Fig4. 16-BIT CSLA With D-Latch Architecture.

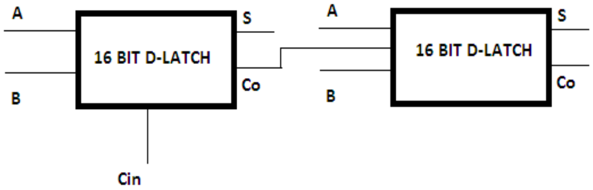


Fig5. 32-BIT CSLA with D-Latch Architecture.

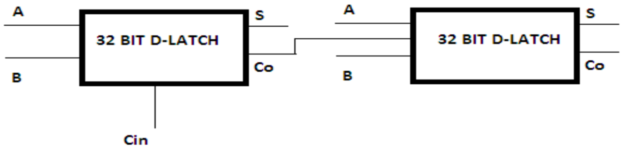


Fig6. 64-BIT CSLA With D-Latch Architecture.

The 16-Bit, 32-Bit and 64-Bit CSLA with D-LATCH architectures were made known in figure 3, figure 4 and 5. The Architecture of 64-Bit CSLA with D-LATCH has been designed based upon cascading of two 32-Bit Architectures. In the D-LATCH architecture first stage was designed based on Ripple Carry adders and the second stage was designed based on D-LATCH logic.

IV. RESULTS

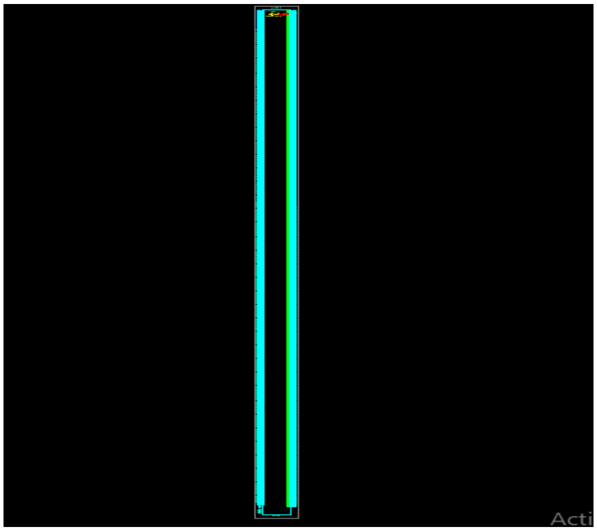


Fig7. RTL Schematic.

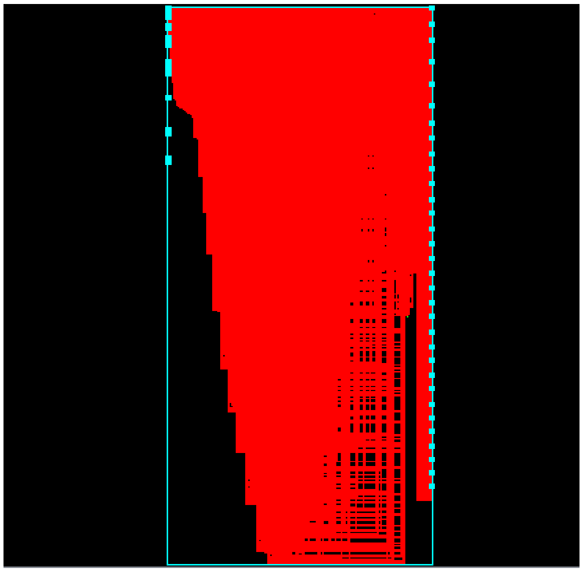
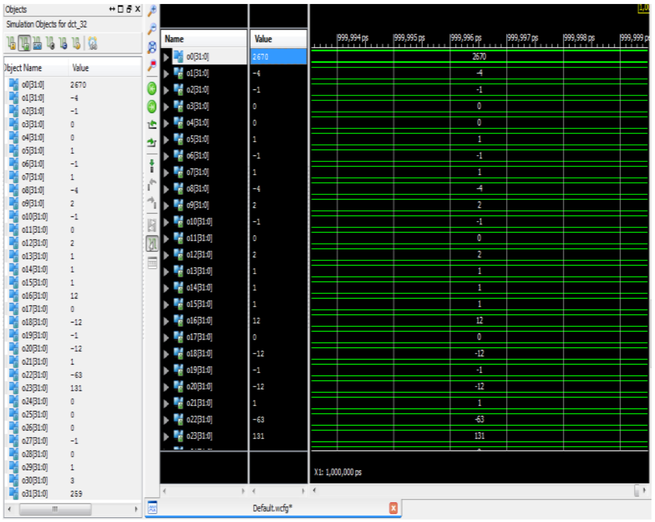


Fig8. Technology Schematic.



Object Name	Value
o0[31:0]	2670
o1[31:0]	-4
o2[31:0]	-1
o3[31:0]	0
o4[31:0]	0
o5[31:0]	1
o6[31:0]	-1
o7[31:0]	1
o8[31:0]	-4
o9[31:0]	2
o10[31:0]	-1
o11[31:0]	0
o12[31:0]	2
o13[31:0]	1
o14[31:0]	1
o15[31:0]	1
o16[31:0]	12
o17[31:0]	0
o18[31:0]	-12
o19[31:0]	-1
o20[31:0]	-12
o21[31:0]	1
o22[31:0]	-63
o23[31:0]	131
o24[31:0]	0
o25[31:0]	0
o26[31:0]	0
o27[31:0]	-1
o28[31:0]	0
o29[31:0]	1
o30[31:0]	3
o31[31:0]	259

Fig9. Simulation Waveform.

V. CONCLUSION

In this paper, we have proposed carry select adder logic for obtaining orthogonal approximation of DCT by using recursive algorithm and the derived DCT pairs of lengths **N/2** instead of **N** additions for input processing. The proposed carry select logic for the approximated DCT has several advantages, such as of regularity, structural simplicity, lower-computational complexity, and scalability. Along with these we have another advantage that is latency (delay) is reduced compared to the previous addition techniques. We have also proposed approximate DCT computation where DCT's of 256-point or 128-point DCTs. This can also be extended to N point DCT by using N /2 point DCT, with reduced latency.

VI. REFERENCES

[1]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, 2006.

[2] C.Loeffler, A.Lightenberg, and G.S.Moschytz, “Practical fast 1-DDCT algorithm with 11 multiplications,” in Proc. Int. Conf. Acoust.,Speech, Signal Process. (ICASSP), May 1989, pp. 988–991.

[3] M. Jridi, P. K. Meher, and A. Alfalou, “Zero-quantised discrete cosine transform coefficients prediction technique for intra-frame video encoding,” IET Image Process., vol. 7, no. 2, pp. 165–173, Mar. 2013.

[4] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, “Binary discrete cosine and Hartley transforms,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 60, no. 4, pp. 989–1002, Apr. 2013.

[5] F. M. Bayer and R. J. Cintra, “DCT-like transform for image compression requires 14 additions only,” Electron. Lett., vol. 48, no. 15, pp. 919–921, Jul. 2012.

[6] R. J. Cintra and F. M. Bayer, “A DCT approximation for image compression,” IEEE Signal Process. Lett., vol. 18, no. 10, pp. 579–582, Oct. 2011.

[7] S. Bouguezel, M. Ahmad, and M. N. S. Swamy, “Low-complexity 8x 8 transform for image compression,” Electron. Lett., vol. 44, no. 21, pp. 1249–1250, Oct. 2008.

[8] T. I. Haweel, “A new square wave transform based on the DCT,” Signal Process., vol. 81, no. 11, pp. 2309–2319, Nov. 2001.

[9] V. Britanak, P.Y.Yip, and K. R. Rao, Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations. London, U.K.: Academic, 2007.

[10] G. J. Sullivan, J.-R. Ohm,W.-J.Han, and T.Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649–1668, Dec. 2012.