IPHV8I2044X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .II
ISSN Online: 2319 – 9253
Print: 2319 – 9245

AN EFFICIENT 16 POINT RECONFIGURABLE DCT ARCHITECTURE USING IMPROVED 8 POINT DCT

[#1]S.NAGA JYOTHI, M.Tech Student,
[#2]A.LAVANYA, Assistant Professor,
DEPT. OF ECE,
DRK INSTITUTE OF SCIENCE AND TECHNOLOGY, BOWRAMPET ,TS, INDIA.

Abstract: In this paper we present a reconfigurable 16 Approximation of discrete cosine transform (DCT) using 8 point additions with less number of calculations. Most of the existing algorithms for approximation of the DCT target only the DCT of small transform lengths, and some of them are non-orthogonal. This paper presents a generalized recursive algorithm to obtain orthogonal approximation of DCT where an approximate DCT of length could be derived from a pair of DCTs of length at the cost of additions for input preprocessing. We perform recursive sparse matrix decomposition and make use of the symmetries of DCT basis vectors for deriving the proposed approximation algorithm. The DCT is employed in a multitude of compression standards due to its remarkable energy compaction properties. Multiplier-free approximate DCT transforms have been proposed that offer superior compression performance at very low circuit complexity. Such approximations can be realized in digital VLSI hardware using additions and subtractions only, leading to significant reductions in chip area and power consumption compared to conventional DCTs and integer transforms.

Key words: Approximate DCT, low-complexity algorithms, Algorithm-architecture co design, DCT approximation, discrete cosine transform (DCT).

I. INTRODUCTION

T HE DISCRETE cosine transform (DCT) is popularly used in image and video compression. Since the DCT is computationally intensive, several algorithms have been proposed in the literature to compute it efficiently. Recently, significant work has been done to derive approximate of 8-point DCT for reducing the computational complexity. The main objective of the approximation algorithms is to get rid of multiplications which consume most of the power and computation-time, and to obtain meaningful estimation of DCT as well. The need of approximation is more important for higher-size DCT since the computational complexity of the DCT grows nonlinearly. On the other hand, modern video coding standards such as high efficiency video coding (HEVC) uses DCT of larger block sizes (up to 32 32) in order to achieve higher compression ratio. But, the extension of the design strategy used in H264 AVC for larger transform sizes, such as 16-point and 32-point is not possible [11]. Besides, several image processing applications such as tracking and simultaneous compression and encryption [13] require higher DCT sizes. In this context, Cintra has introduced a new class of integer transforms applicable to several block-lengths. Cintra et al. have proposed a new 16 16 matrix also for approximation of 16-point DCT, and have validated it experimentally. Recently, two new transforms have been proposed for 8-point DCT approximation: Cintra et al. have proposed a low-complexity 8-point approximate DCT based on integer functions and Potluri et al. have proposed a novel 8-point DCT approximation that requires only 14 additions. On the other hand, Bouguezel et al. have proposed two methods for multiplication-free approximate form of DCT. The first method is for length, 16 and 32; and is based on the appropriate extension of integer DCT [18]. Also, a systematic method for developing a binary version of high-size DCT (BDCT) by using the sequency-ordered Walsh-Hadamard transform (SO-WHT) is proposed in [4]. This transform is a permutated version of the WHT which approximates the DCT very well and maintains all the advantages of the WHT.

A scheme of approximation of DCT should have the following features:

i) It should have low computational complexity.
ii) It should have low error energy in order to provide compression performance close to the exact DCT, and preferably should be orthogonal.

But the existing DCT algorithms do not provide the best of all the above three requirements. Some of the existing methods are deficient in terms of scalability [18], generalization for higher sizes [15], and orthogonality. We

intend to maintain orthogonality in the approximate DCT for two reasons. Firstly, if the transform is orthogonal, we can always find its inverse, and the kernel matrix of the inverse transform is obtained by just transposing the kernel matrix of the forward transform. This feature of inverse transform could be used to compute the forward and inverse DCT by similar computing structures. Moreover, in case of orthogonal transforms, similar fast algorithms are applicable to both forward and inverse transforms.

In this context, the discrete cosine transform (DCT) is an essential mathematical tool in both image and video coding. Indeed, the DCT was demonstrated to provide good energy compaction for natural images, which can be described by first-order Markov signals . Moreover, in many situations, the DCT is a very close substitute for the Karhunen-Loève transform (KLT), which has optimal properties. As a result, the two-dimensional (2-D) version of the 8-point DCT was adopted in several imaging standards such as Additionally, new compression schemes such as the High Efficiency Video Coding (HEVC) employs DCT-like integer transforms operating at various block sizes ranging from 4 4 to 32 32 pixels . The distinctive characteristic of HEVC is its capability of achieving high compression performance at approximately half the bit rate required by H.264/AVC with same image quality. Also HEVC was demonstrated to be especially effective for high-resolution video applications. Therefore, low complexity DCT-like approximations may benefit future video codecs including emerging HEVC/H.265 systems. Several efficient algorithms were developed and a noticeable literature is available. Although fast algorithms can significantly reduce the computational complexity of computing the DCT, floating-point operations are still required. Despite their accuracy, floating-point operations are expensive in terms of circuitry complexity and power consumption. Therefore, minimizing the number of floating-point operations is a sought property in a fast algorithm. One way of circumventing this issue is by means of approximate transforms

II. EXISTING METHOD

In current literature, several approximate methods for the DCT calculation have been archived [11]. While not computing the DCT exactly, such approximations can provide meaningful estimations at low-complexity requirements. In particular, some DCT approximations can totally eliminate the requirement for floating-point operations—all calculations are performed over a fixed-point arithmetic framework. Prominent 8-point approximation-based techniques.

In general, these approximation methods employ a transformation matrix whose elements are defined over the set . This implies null multiplicative complexity, because the required operations can be implemented exclusively by means of binary additions and shift operations. Such DCT approximations can provide low-cost and low-power designs and effectively replace the exact DCT and other DCT-like transforms. Indeed, the performance characteristics of the low complexity DCT approximations appear similar to the exact DCT, while their associate hardware implementations are economical because of the absence of multipliers [14]. As a consequence, some prospective applications of DCT approximations are found in real-time video transmission and processing.

Emerging video standards such as HEVC provide for reconfigurable operation on-the-fly which makes the availability of an ensemble of fast algorithms and digital VLSI architectures a valuable asset for low-energy high-performance embedded systems. For certain applications, low circuit complexity and/or power consumption is the driving factor, while for certain other applications, highest picture quality for reasonably low power consumption and/or complexity may be more important. In emerging systems, it may be possible to switch modus operandi based on the demanded picture quality vs available energy in the device. Such feature would be invaluable in high quality smart video devices demanding extended battery life. Thus, the availability of a suite of fast algorithms and implementation libraries for several efficient DCT approximation algorithms may be a welcoming contribution

Furthermore, another possible application for a suite of DCT approximation algorithms in the light of reconfigurable video codec's is the intelligent intra-frame fast reconfiguration of the DCT core to take into account certain local frame information and measured SNR metrics. For example, certain parts of a frame can demand better picture quality (foreground, say) when compared to relatively unimportant part of the frame (background, say) . In such a case, it may be possible to switch DCT approximations algorithms on an intra frame basis to take into account the varying demands for picture clarity within a frame as well as the availability of reconfigurable logic based digital DCT engines that support fast reconfiguration in real-time.

we review the mathematical description of the selected 8-point DCT approximations. All discussed methods here consist of a transformation matrix that can be put in the following format:

We aim at deriving a novel low-complexity approximate DCT. For such end, we propose a search over the 8 8 matrix space in order to find candidate matrices that possess low computation cost. Let us define the cost of a transformation matrix as the number of arithmetic operations required for

its computation. One way to guarantee good candidates is to restrict the search to matrices whose entries do not require multiplication operations. Thus we have the following optimization problem:

We propose digital computer architectures that are custom designed for the real-time implementation of the fast algorithms described in Section II. The proposed architectures employs two parallel realizations of DCT approximation blocks, as shown in Fig. 1. The 1-D approximate DCT blocks implement a particular fast algorithm chosen from the collection described earlier in the paper. The row- and column-wise transforms can be any of the DCT approximations detailed in the paper. In other words, there is no restriction for both row- and column-wise transforms to be the same. However, for simplicity, we adopted identical transforms for both steps

Between the approximate DCT blocks a real-time row-parallel transposition buffer circuit is required. Such block ensures data ordering for converting the row-transformed data from the first DCT approximation circuit to a transposed format as required by the column transform circuit. The transposition buffer block is detailed in Fig. 2.

The digital architectures of the discussed approximate DCT algorithms were given hardware signal flow diagrams as listed below:

1) Proposed novel algorithm and architecture shown in Fig. 3(a);

2) BAS-2008 architecture shown in Fig. 3(b);

3) BAS-2011 architecture shown in Fig. 3(c);

4) CB-2011 architecture shown in Fig. 3(d);

5) Modified CB-2011 architecture shown in Fig. 3(e);

6) Architecture for the algorithm in [40] shown in Fig. 3(f)

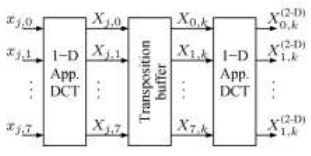

Fig. 1. Two-dimensional approximate transform by means of 1-D approximate transform.

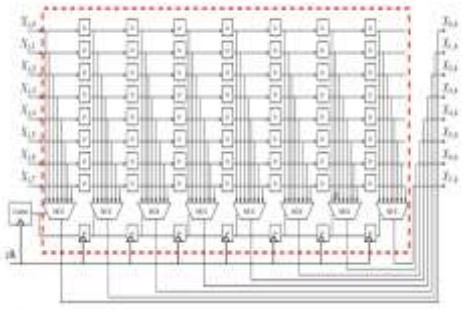

Fig. 2. Details of the transposition buffer block.

Fig. 3. Digital architecture for considered DCT approximations. (a) Proposed approximate transform , (b) BAS-2008 approximate DCT , (c) BAS-2011 approximate DCT where , (d) CB-2011 approximate DCT , (e) Modified CB-2011 approximate DCT , (f) Approximate DCT in [40] .

III. PROPOSED METHOD

3.1 Proposed Scalable Design

The basic computational block of algorithm for the proposed DCT approximation, is given in [6]. The block diagram of the computation of DCT based on is shown in Fig. 1. For a given input sequence , , the approximate DCT coefficients are obtained by . An example of the block diagram of is illustrated in Fig. 2, where two units for the computation of are used along with an input adder unit and output

permutation unit. The functions of these two blocks are shown respectively in (8) and (6). Note that structures of 16-point DCT of Fig. 2 could be extended to obtain the DCT of higher sizes. For example, the structure for the computation of 32-point DCT could be obtained by combining a pair of 16-point DCTs with an input adder block and output permutation block

3.2 Complexity Comparison

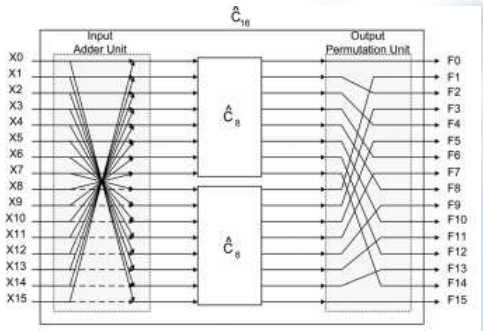

Fig. 4. Block diagram of the proposed DCT for N=16.

N	Method	Arithmetic operations	
		Add	Shift
8	Proposed=BC [6]	22	0
	BDCT [4]	24	0
	BAS [18]	24	4
	BC [6]	22	0
16	Proposed	60	0
	BDCT [4]	64	0
	BAS [18]	64	8
	BC [15]	72	0
32	Proposed	152	0
	BDCT [4]	160	0
	BAS [18]	160	16
64	Proposed	368	0
	BDCT [4]	384	0

TABLE I ASSESSMENT OF REQUIRED ARITHMETIC OPERATIONS FOR SEVERAL APPROXIMATION ALGORITH

and 64-point DCT approximations are 60, 152, and 368 additions, respectively. More generally, the arithmetic complexity of N -point DCT is equal to $N(\log_2 N-(1/4))$ additions. Moreover, since the structures for the computation of DCT of different lengths are regular and scalable, the computational time for N DCT coefficients can be found to be $\log_2(N)Ta$ where is the addition-time. The number of arithmetic operations involved in proposed DCT approximation of different lengths and those of the existing competing approximations are shown in Table I. It can be found that the proposed method requires the lowest number of additions, and does not require any shift operations. Note that shift operation does not involve any combinational components, and requires only rewiring during hardware implementation. But it has indirect contribution to the 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 operation. Also, we note that all considered approximation methods involve significantly less computational complexity over that of the exact DCT algorithms. According to the Loeffler algorithm [2], the exact DCT computation requires 29, 81, 209, and 513 additions along with 11, 31, 79, and 191 multiplications, respectively for 8, 16, 32, and 64-point DCTs.

Pipelined and non-pipelined designs of different methods are developed, synthesized and validated using an integrated logic analyzer. The validation is carried out by using the Digilent EB of Spartan6-LX45. We have used 8-bit inputs, and we have allowed the increase of output size (without any truncations). For the 8-point transform of Fig. 1, we have 11-bit and 10-bit outputs. The pipelined design are obtained by insertion of registers in the input and output stages along with registers after each adder stage, while the no pipeline registers are used within the non-pipelined designs. The synthesis results obtained from XST synthesizer are presented in Table II. It shows that pipelined designs provide significantly higher maximum operating frequency (MOF). It also shows that the proposed design involves nearly 7%, 6%, and 5% less area compared to the BDCT design for equal to 16, 32, and 64, respectively. Note that both pipelined and non-pipelined designs involve the same number of LUTs since pipeline registers do not require additional LUTs. For 8-point DCT, we have used the approximation proposed in [6] which forms the basic computing block of the proposed method. Also, we underline that all designs have the same critical path; and accordingly have the same MOFs. Most importantly, the proposed designs are reusable for different transform lengths.

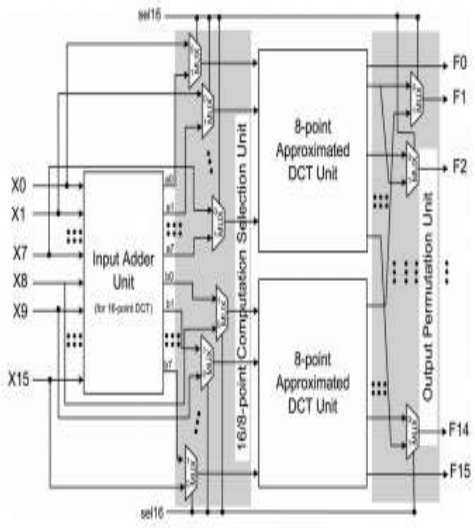

Fig. 5. Proposed reconfigurable architecture for approximate DCT of lengths n=8 and 16.

1. **Output wave forms**

Fig 6:output waveform

As we say that we are performing 16 point reconfigurable using no multiplications ,it depends on sel line if sel goes low 8 point dct is performed or else 16 point dct is performed

IV.CONCLUSION

we have proposed a recursive algorithm to obtain orthogonal approximation of DCT where approximate DCT of length 16 could be derived from a pair of DCTs of length 8 at the cost of additions 16 for input preprocessing. The proposed approximated DCT has several advantages, such as of regularity, structural simplicity, lower-computational complexity, and scalability. Comparison with recently proposed competing methods shows the effectiveness of the proposed approximation in terms of error energy, hardware resources consumption, and compressed image quality

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-D DCT 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]Narasimha, M.; Peterson, A. (June 1978). "On the Computation of the Discrete Cosine Transform". IEEE Transactions on Communications. **26** (6): 934–936.doi:10.1109/TCOM.1978.1094144.

[8]Makhoul, J. (February 1980). "A fast cosine transform in one and two dimensions". IEEE Transactions on Acoustics, Speech, and Signal Processing. **28** (1): 27–34.doi:10.1109/TASSP.1980.1163351.

[9]Sorensen, H.; Jones, D.; Heideman, M.; Burrus, C. (June 1987). "Real-valued fast Fourier transform algorithms". IEEE Transactions on Acoustics, Speech, and Signal Processing. **35** (6): 849–863. doi:10.1109/TASSP.1987.1165220.

[10]Arai, Y.; Agui, T.; Nakajima, M. (November 1988). "A fast DCT-SQ scheme for images". IEICE Transactions. **71** (11): 1095–1097.

[11]Plonka, G.; Tasche, M. (January 2005). "Fast and numerically stable algorithms for discrete cosine transforms". Linear Algebra and its Applications. **394** (1): 309–345.doi:10.1016/j.laa.2004.07.015.

[12]Duhamel, P.; Vetterli, M. (April 1990). "Fast fourier transforms: A tutorial review and a state of the art". Signal Processing. **19** (4): 259–299. doi:10.1016/0165-1684(90)90158-U.

AUTHOR'S PROFILE

[1].S.NAGA JYOTHI, Now Studying M.Tech in VLSI and Embedded Systems stream in Department of Electronics and Communication Engineering in DRK Institute of Science & Technology Bowrampet(V),Telangana, India.

[2]. A.LAVANYA Presently Working as Assistant Professor in Department of ECE from DRK Institute of Science & Technology, Bowrampet (V), Hyderabad, Telangana, India.