International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 05
April 2017

Reconfigurable Architecture for Efficient and Scalable Orthogonal Approximation of DCT Using Verilog HDL

B. LAXMAN
Assistant Professor, Department of ECE
Chaitanya Institute of Technology and Science, Warangal, Telangana

Abstract: A high speed word level finite field multiplier in F2 m using redundant representation is proposed. For the class of finite field that there exists a type I optimal normal basis, the new architecture has sign can higher speed compared to previously proposed architectures using either normal basis or redundant representation at the expense of moderately higher area complexity . One of the unique features of the proposed multiplier is that the critical path delay is not a function of the eld size nor the word size. It is shown that the new multiplier out-performs all the other multipliers in comparison when considering the product of area and delay as a measure of performance. VLSI implementation of the proposed multiplier in a 0:18m CMOS process is also presented.

I Introduction

Finite fields have important applications in error control coding and cryptography [1]. Finite fields of characteristic two are most commonly used because they are inherently suitable for VLSI implementation. In practice, finite field multiplier is the key arithmetic unit for many systems based on finite field computations since more complicated finite field operations such like division and in version can be broken down into a series of consecutive multiplication operations. Efficiency of finite field multiplication depends on the choice of the basis to represent finite field elements.

Bases that have been used for realizing finite field multipliers include polynomial basis, normal basis (NB), dual bases, triangular basis, redundant representation or redundant basis, and their variations (i.e., shifted polynomial basis) [2], [3], [7], [8], [9], [10], [11]. Redundant representation is especially interesting because it not only offers almost free squaring as NB does but also eliminates modular operation for multiplication. The main drawback for the redundant representation is that it uses more bits to represent a finite field element, where the number of representation bits depends on the size of the cyclostomes ring in which the underlying eld is embedded.

Efficient computations in finite fields and their architectures are important in many applications including coding theory, computer algebra systems and public-key cryptosystems (e.g., elliptic curve cryptosystems). Although all finite fields of the same cardinality are isomorphic, their arithmetic efficiency depends greatly on the choice of bases for field element representations. The most commonly used bases are polynomial bases (PB) and normal bases (NB), sometimes combined with dual bases (DB)[15]. A major advantage of normal bases in the fields of characteristic two is that the squaring operation inNBis simply a cyclic shift of the coordinates of elements, so these are useful for computing large exponentiations and multiplicative inverses [13, 11, 1]. Also, the multiplication table of a normal basis is symmetric, so suitable for hardware implementation.

We are mainly interested in finite fields of characteristic two, i.e.F2m, which are one of the two types of fields used most commonly in

practice (the other one is Fp where p is a prime). We show how to find the smallest cyclostomes ring in which F2 m can be embedded. Since "embedding" is not unique, each element in the ring can be represented in more than one way, i.e.,the representation contains certain amount of redundancy. In this article, we also discuss how this redundant representation of a field element can be efficiently converted to a normal basis and vice versa.

II.Literature Survey

A representation of finite fields that has proved useful when implementing finite field arithmetic in hardware is based on an isomorphism between subrings and fields. In this paper, we present an unified formulation for multiplication in cyclotomic rings and cyclotomic fields in that most arithmetic operations are done on vectors. From this formulation we can generate optimized algorithms for multiplication. F or example, one of the proposed algorithms requires approximately half the number of coordinate-level multiplications at the expense of extra coordinate-level additions. Our method is then applied to the finite fields GF(q m) to further reduce the number of operations. We then present optimized algorithms for multiplication in finite fields with type-I and type-II optimal normal bases.

This article presents simple and highly regular architectures for finite field multipliers using a redundant representation. The basic idea is to embed a finite field into a cyclotomic ring which has a basis with the elegant multiplicative structure of a cyclic group. One important feature of our architectures is that they provide area-time trade-offs which enable us to implement the multipliers in a partial-parallel/hybrid fashion. This hybrid architecture has great significance in its VLSI implementation in very large fields. The squaring operation using the redundant representation is simply a permutation of the coordinates. It is shown that when there is an optimal normal basis, the proposed bit-serial and hybrid multiplier architectures have very low space complexity. Constant multiplication is also considered and is shown to have advantage in using the redundant representation. The elements of -point DCT matrix are given by:

A new GF (2 n) redundant representation is presented. Squaring in the representation is almost cost-free. Based on the representation, two multipliers are proposed. The XOR gate complexity of the first multiplier is lower than a recently proposed normal basis multiplier when CN (the complexity of the basis) is larger than 3n-1.

III. Scalable and Reconfigurable Architecture for DCT Computation

we discuss the proposed scalable architecture for the computation of approximate DCT of and 32. We have derived the theoretical estimate of its hardware complexity and discuss the recon figuration scheme.

A. 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 $\{X(n)\}, n \in [0, N-1]$, the approximate DCT coefficients are obtained by $F=C_N^{\wedge}.X^t$.

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.

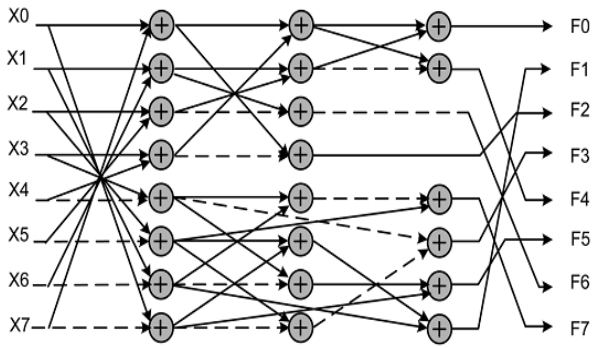


Fig. 1. Signal flow graph (SFG) of (C^8). Dashed arrows represent multiplications by 1

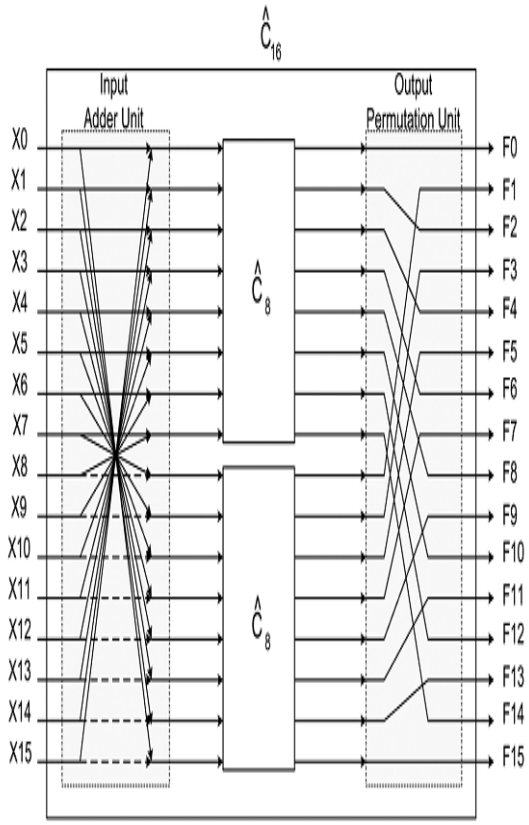


FIG.2.block diagram of proposed DCT for N=16(c^16)

B.Complexity Comparison

To assess the computational complexity of proposed –point approximate DCT, we need to determine the computational cost of matrices quoted in (9). As shown in Fig. 1 the approximate 8-point DCT involves 22 additions. Since has no computational cost and requires additions for – point DCT, the overall arithmetic complexity of 16-point, 32-point,And 64-point DCT approximations are 60, 152, and 368 additions, respectively. More generally, the arithmetic complexity of -point DCT is equal to additions.

C.Proposed reconfiguration scheme

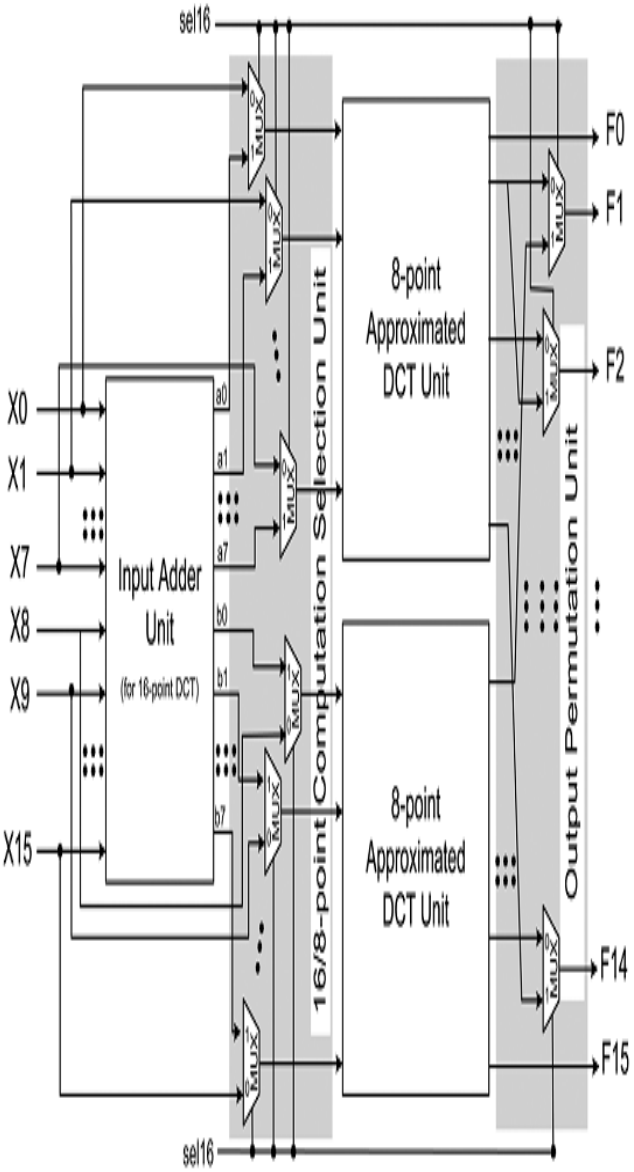


Fig. 3. Proposed recon figurable architecture for approximate DCT of lengths N=8 and 16

As specified in the recently adopted HEVC [10], DCT of different lengths such as, 16,32 are required to be used in video coding applications. Therefore, a given DCT architecture should be potentially reused for the DCT of different lengths instead of using separate structures for different lengths. We propose here such reconfigurable DCT structures which could be reused for the computation of DCT of different lengths. The reconfigurable architecture for the implementation of approximated 16-point DCT is shown in Fig. 3. It consists of three computing units, namely two 8-point approximated DCT units and a 16-point input adder unit that generates a(i) and b(i) he input to the first 8-point DCT approximation unit is fed through 8 MUXes that select either[a(0)…a(7)] or ,[x(0)….x(7)] depending on whether it is used for 16-point DCT calculation or 8-point DCT calculation. Similarly, the input to the second 8-point DCT unit (Fig. 3) is fed through 8 MUXes that select either [b(0)…b(7)] or , depending on whether it is used for 16-point DCT calculation or 8-point DCT calculation. On the other hand, the output permutation unit uses 14 MUXes to select and re-order the output depending on the size of the selected DCT. is used as control input of the MUXes to select inputs and to perform permutation according to the size of the DCT to be computed. Specifically sel16=1 enables the computation of 16-point DCT and sel16=0 enables the computation of a pair of 8-point DCTs in parallel. Consequently, the architecture of Fig. 3 allows the calculation of a 16-point DCT or two 8-point DCTs in parallel.

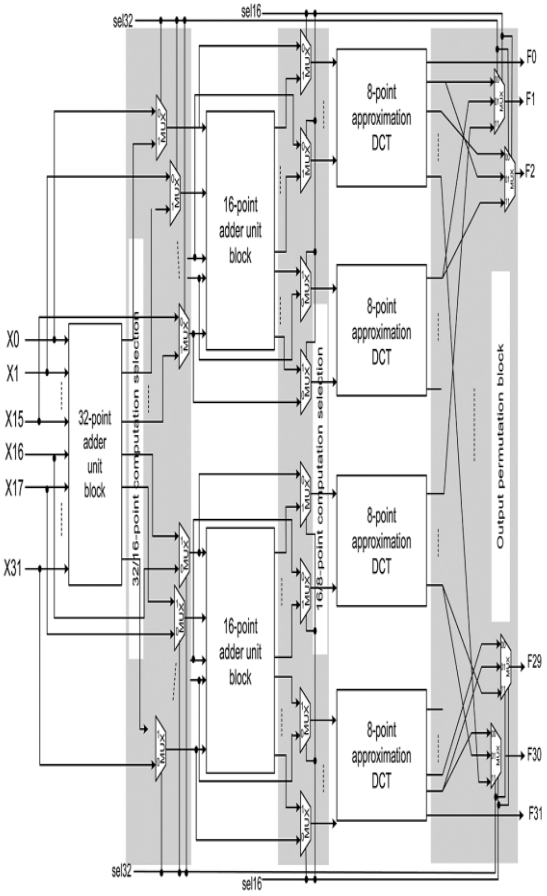


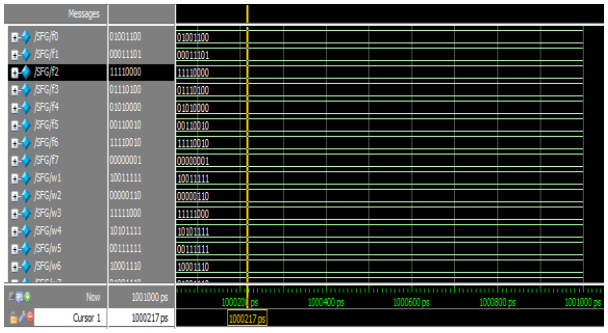
Fig. 4. Proposed recon figurable architecture for approximate DCT of lengths , 16 and *32*

A reconfigurable design for the computation of 32-, 16-, and 8-point DCTs is presented in Fig. 4. It performs the calculation of a 32-point DCT or two 16-point DCTs in parallel or four 8-point DCTs in parallel. The architecture is composed of 32-point input adder unit, two 16-point input adder units, and four 8-point DCT units. The reconfigurability is achieved by three control blocks composed of 64 2:1 MUXes along with 30 3:1 MUXes. The first control block decides whether the DCT size is of 32 or lower. If , the selection of input data is done for the 32point DCT, otherwise, for the DCTs of lower lengths.

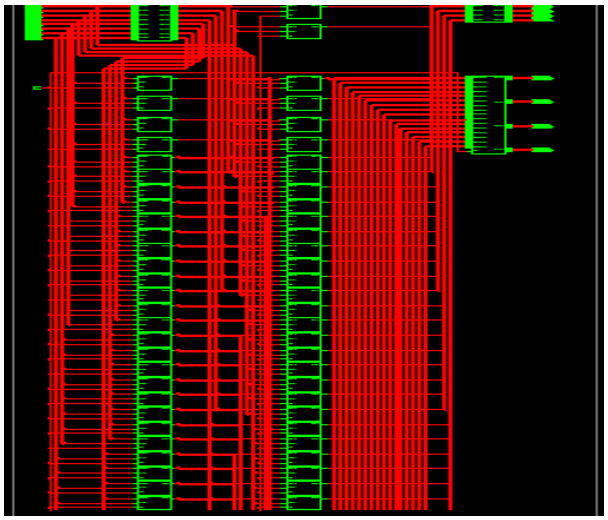
International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 05
April 2017

Simulation Results:

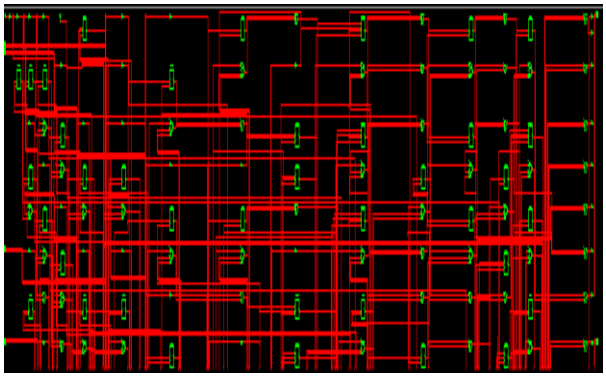
Top Module:



RTL Schematic:



Technology schematic:



Design summary:

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	276	14752	1%
Number of 4 input LUTs	509	29504	1%
Number of bonded IOBs	129	250	51%
Number of MULT18X18SIOs	4	36	11%

IV Conclusion

A reconfigurable bit-serial Galois field multiplier architecture is proposed in this paper. The multiplier is reconfigurable because it can perform for variable Galois field degreem: This multiplier can support any arbitrary irreducible polynomial. The multiplication result is computed aftermclock cycles. The advantages of the proposed architecture are the high order of flexibility, which allows an easy configuration for variable field size 2m ; and the low hardware complexity, which results in small area. In addition, the proposed multiplier has low power consumption features, which are achieved by using the gated clock technique. Comparing with previous published implementations, the proposed multiplier architecture is suitable for devices with limited silicon area.

Extension Work:

In Proposed system we are using Input Adder Unit, now it can be replaced by Wallace Tree Multiplier. By doing this we can get less power consumption, high accuracy and reduced delay.

V References

[1] A. M. Shams, A. Chidanandan, W. Pan, and M. A. Bayoumi, "NEDA: A low-power high-performance DCT architecture," I E E E 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,"Binarydiscrete 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 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,andK.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.

[11] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation analysis," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1685–1696, 2012.

[12] X.Li,A.Dick,C.Shen,A.vandenHengel,andH. Wang, "Incremental learning of 3D-DCT compactrepresentations for robust visual tracking," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 4, pp. 863–881, Apr. 2013.

[13] A. Alfalou, C. Brosseau, N. Abdallah, and M. Jridi, "Assessing the performance of a method of simultaneous compression and encryption of multiple images and its resistance against various attacks,"Opt. Express, vol. 21, no. 7, pp. 8025–8043, 2013.

[14] R. J. Cintra, "An integer approximation method for discrete sinusoidal transforms," Circuits, Syst., Signal Process., vol. 30, no. 6, pp. 1481–1501, 2011.

[15] F. M. Bayer, R. J. Cintra, A. Edirisuriya, and A. Madanayake, "A digital hardwarefast algorithm and FPGA-based prototype for a novel 16-point approximate DCT for image compression applications," Meas. Sci. Technol., vol. 23, no. 11, pp. 1–10, 2012.

[16] R. J. Cintra,F. M. Bayer, and C. J. Tablada, "Low-complexity 8-point DCT approximations based on integer functions,"Signal Process., vol. 99, pp. 201–214, 2014.