Available online at www.elixirpublishers.com (Elixir International Journal)

Electrical Engineering

Elixir Elec. Engg. 56A (2013) 13769-13771

ISSN: 2229-712X

Implementation of reconfigurable IDCT architecture on FPGA for multiple video standards

K.Swaminathan and Sheeba

Department of Electronics and Communication Engineering, J.J. College of Engineering and Technology, Tiruchirappalli, Tamilnadu, India.

ARTICLE INFO

Article history:
Received: 13 August 2012;
Received in revised form:
21 March 2013;
Accepted: 25 March 2013;

Keywords
Circuit share,
High-definition video,
Multistandard inverse transform,
Multi-standard,
IDCT,
Reconfigurable architecture.

ABSTRACT

In this brief, a reconfigurable IDCT architecture is designed for multistandard inverse transform. The proposed architecture is used in multistandard decoder of MPEG-2, MPEG-4 ASP, H.264/AVC and VC-1. Two circuits share strategies, factor share (FS) and adder share (AS), are applied to the inverse transform architecture for saving its circuit resource.

Introduction

Recently, reconfigurable concept has become one of the most important issues for video coding technology. People usually hope that a product can support various formats. From manufacturer and researcher view, how to development the reconfigurable architecture or algorithm is an important issue. There are many video standards such as MPEG-1/2/4, VC-1 and H.264/AVC. In intra-frame decoding, they both have IDCT transform. The transform types are different in various video standards. MPEG-1/2/4 has 8x8 floating-point transform, H.264/AVC has 8x8 integer transform and 4x4 integer/Hadamard transform; moreover, VC-1 has 8x8, 8x4, 4x8 and 4x4 integer transform. Because of the transform types are different; the integration of multistandard codec does not mean that several standard codec's are simply gathered together. It is expected to have a higher density of integration by circuit share. Many coding tools from different compression standards are similar, even though their detail algorithms are different, such as motion estimation, inverse transform and inverse quantization, and variable length decoding. Similar coding tools from different standards may be efficiently integrated in a single chip through elaborating circuit share, so that the area of the integrated multistandard chip is much smaller than the total areas of these single standard chips. Because of the transform types are different; we proposed a reconfigurable IDCT architecture that can solve this issue

Discrete cosine transform (DCT) is a key coding tool for video compression. It achieves data compression by converting the high relative spatial domain data into low relative frequent domain data. It was first introduced in image coding by Rao et al. [1]. The DCT and inverse DCT (IDCT) are employed in all video coding standards mentioned above. The similarities of DCT/IDCT from different coding standards may be shared, which is beneficial for reducing the cost of very large scale integration (VLSI) implementation of multistandard DCT/IDCT architecture. Moreover, due to the fact that DCT/IDCT units are frequently called in the video codec loop, its performance has an important influence on the overall performance of the codec. In [2], the delta matrix is employed for sharing the circuits. In [3], an application-specific instruction set processor controlled inverse transform is proposed for high design flexibility. A high parallel architecture is proposed for all transforms of H.264/AVC in [4] and in [5], the matrix decomposition is used in inverse transform architecture for circuit saving. All these mentioned transform architectures aim at achieving high performance and low cost.

The remainder of this brief is organized as follows. The matrix decomposition of DCT and IDCT from MPEG-2/4,H.264/AVC, and VC-1 are reviewed in Section II. The proposed optimization strategies for improving the VLSI architecture of the multistandard IDCT, factor share (FS) and adder share (AS),are introduced in Section III. The VLSI architecture is designed in Section IV and its synthesis results are shown in Section V. Finally, Section VI concludes this brief.

Algorithm Of Discrete Cosine Transform

In video compression standards, transform coding usually employs 8-point or 4-point II-type DCTs [6]. The one-dimensional (1-D) 8-point IDCT *T*8 can be expressed in matrix form as follows:

$$T_8=\begin{bmatrix} a & b & f & c & a & d & g & e \\ a & c & g & -e & -a & -b & -f & -d \\ a & d & -g & -b & -a & e & f & c \\ a & e & -f & -d & a & c & -g & -b \\ -a & -e & f & d & -a & -c & g & b \\ a & -d & -g & -b & -a & -e & f & -c \\ a & -c & g & -e & -a & b & -f & d \end{bmatrix}$$

Tele:
E-mail addresses: swaminathan.vinoth@gmail.com

where a to g is DCT coefficient having integer values depending on the video standards. The implementation of above matrix needs more number of multiplication and addition.

For solving this problem we may use Recursion property of DCT transform. It has been proved that the 2N-point II-type IDCT can be decomposed into two different types of N-point IDCT [6]. Following the recursion property, the 8-point II-type IDCT T8 can be decomposed into a 4-point IDCT and a 4-point IDCT. The decomposition of a 1-D 8-point IDCT is expressed as follows:

$$T8=\begin{bmatrix} T4 & 0 \\ 0 & V4 \end{bmatrix}$$

Where T4 and V4 are given by

$$T4=\begin{bmatrix} a & f & a & g \\ a & g & -a & -f \\ a & -g & -a & f \\ a & -f & a & -g \end{bmatrix}$$

$$V4=\begin{bmatrix} e & d & c & -b \\ d & b & e & c \\ c & e & -b & -d \\ b & -c & d & e \end{bmatrix}$$

The transform of a 1-D IDCT is expressed as follows:

$$T8X8=\begin{bmatrix} T4 & 0 \\ 0 & V4 \end{bmatrix}$$

Where

$$X=\begin{bmatrix} X0 & X1 X2 X3 & X4 & X5 & X6 & X7 & X8 \end{bmatrix}^T$$

Then, the variable input is separated into odd and even terms and is multiply with V4, T4 matrixes as shown as below:

$$T4=\begin{bmatrix} a & f & a & g \\ a & g & -a & -f \\ a & -g & -a & f \\ a & -f & a & -g \end{bmatrix}\begin{bmatrix} X0 \\ X2 \\ X4 \\ X6 \end{bmatrix}$$

$$V4=\begin{bmatrix} e & d & c & -b \\ d & b & e & c \\ c & e & -b & -d \\ b & -c & d & e \end{bmatrix}\begin{bmatrix} X1 \\ X3 \\ X5 \\ X7 \end{bmatrix}$$

The implementation of T4 or V4 needs 16 multiplications and 12 additions. The implementation of butterfly matrix needs 8 additions. In total, the decomposition implementation of a 1-D 8-point IDCT needs 32 (16 × 2) multiplications and 32 (12 × 2+ 8) additions.

The elements of the IDCT matrix a ∼ f are real numbers. The real number computation is inappropriate for discrete digital signal processing in practice applications. Thus, substituting for the real IDCT, the integer IDCT is used in video codec’s for simple digital implementation. Integer IDCT is the approximation of the real IDCT. It has negligible performance loss, whereas lower complexity than the real IDCT. Each one of elements a ∼ f is different with different integer IDCTs. Different integer IDCTs are defined in H.264/AVC and VC-1.Their 8-point and 4-point integer IDCT matrices are given as

$$T4avc=\begin{bmatrix} 1 & 1 & 1 & ½ \\ 1 & ½ & -1 & -1 \\ 1 & -½ & -1 & 1 \\ 1 & -1 & 1 & -½ \end{bmatrix}$$

$$T4vc1=\begin{bmatrix} 17 & 22 & 17 & 10 \\ 17 & 10 & -17 & -22 \\ 17 & -10 & -17 & 22 \\ 17 & -22 & 17 & -10 \end{bmatrix}$$

$$T8avc=\begin{bmatrix} 8 & 12 & 8 & 10 & 8 & 6 & 4 & 3 \\ 8 & 10 & 4 & -3 & -8 & -12 & -8 & -6 \\ 8 & 6 & -4 & -12 & -8 & 3 & 8 & 10 \\ 8 & 3 & -8 & -6 & 8 & 10 & -4 & -12 \\ 8 & -3 & -8 & 6 & 8 & -10 & -4 & 12 \\ 8 & -6 & -4 & 12 & -8 & -3 & 8 & -10 \\ 8 & -10 & 4 & 3 & -8 & 12 & -8 & 6 \\ 8 & -12 & 8 & -10 & 8 & -6 & 4 & -3 \end{bmatrix}$$

$$T8vc1=\begin{bmatrix} 12 & 16 & 16 & 15 & 12 & 9 & 6 & 4 \\ 12 & 15 & 6 & -4 & -12 & -16 & -16 & -9 \\ 12 & 9 & -6 & -6 & -12 & 4 & 16 & 15 \\ 12 & 4 & -16 & -9 & 12 & 15 & -6 & -16 \\ 12 & -?4 & -16 & 9 & 12 & -15 & -6 & 16 \\ 12 & -9 & -6 & 16 & -12 & -4 & 16 & -15 \\ 12 & -15 & 6 & 4 & -12 & 16 & -16 & 9 \\ 12 & -16 & 16 & -15 & 12 & -9 & 6 & -4 \end{bmatrix}$$

Circuit Share Strategies

The circuit area can be efficiently reduced by adopting appropriate circuit share strategies. Multiplication operations are needed in traditional IDCT processing. However, the circuit of multiplier is relatively complicated for VLSI implementation. Thus, the multiplier-less transform is preferred. In the multiplier-less transform, each element of the IDCT matrix is equally expressed as the sum of several binary factors. Although some elements in the integer IDCT matrix are different, some sums of their binary factors (SBFs) are possibly the same. The same SBFs can be shared in the multiplier-less implementation of the integer IDCT. This circuit optimization strategy is called as FS. Let E0 and E1 denote two different elements of the IDCT matrix, the binary factorization of E0 and E1 are expressed as

$$E_0=\sum_{i=0}^{N} r_i 2^i, \quad E_1=\sum_{i=0}^{N} s_i 2^i, \quad r_i, s_i=\pm 1, 0, \quad N \geq 0$$

where N is the maximum bitwidth of all integer elements. Extracting the same SBFs eΔm from E0 and E1, then it is rewritten as

$$E_0=\sum_{m=0}^{M-1} 2^{j_m} e_{\Delta m}+e_0$$

$$E_1=\sum_{m=0}^{M-1} 2^{k_m} e_{\Delta m}+e_1$$

$$e_{\Delta m}=\sum_{i=0}^{N} t_i 2^i, \quad t_i=\begin{cases} 0, & r_{i+j_m} \neq s_{i+k_m} \\ r_{i+j_m}, & r_{i+j_m}=s_{i+k_m} \end{cases}$$

$$e_0=\sum_{i=0}^{N} u_i 2^i, \quad u_i=\begin{cases} r_i, & r_i \neq s_i \\ 0, & r_i=s_i \end{cases}$$

$$e_1=\sum_{i=0}^{N} v_i 2^i, \quad v_i=\begin{cases} s_i, & r_i \neq s_i \\ 0, & r_i=s_i \end{cases}$$

It is not necessary to implement two circuits of the same SBFs eΔm for computing E0 and E1, respectively. One circuit of the same SBFs eΔm can be shared in the computations of E0 and E1. 2jmeΔm can be implemented by jm-bits left-shift of eΔm. The circuit comparison of applying and not applying FS in the condition of M = 1 is presented in Fig. 1.

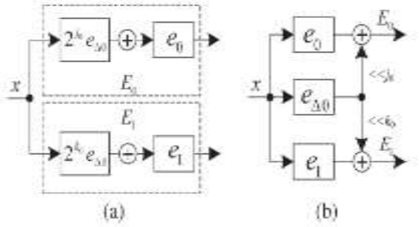


Fig. 1. Circuit of element factorization (a) without FS; (b) with FS.

In this way factor sharing is used to increase circuit utilization. For example applying FS to optimize the circuit of MPEG-2/4 IDCT, 5 adders are saved and only 19 adders are needed to implement the factorized elements. The proposed circuit architecture of the subunit bcde(x), whose function outputs are bx, cx, dx, and ex, is shown in Fig. 2,where the shared adders are highlighted in bold.

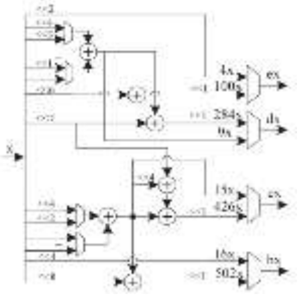


Fig. 2. Proposed circuit architecture of the subunit bcde(x) of integrating MPEG-2/4 and VC-1 8-point IDCTs based on AS.

Vlsi Architecture Design And Implementation

The basic framework of the proposed 8-point 1-D multistandard IDCT architecture is based on (3). All 8-point IDCTs are decomposed into two 4-point IDCTs T4 and V4, a butterfly matrix P8,l, and a permutation matrix P8,r. The computations of 1 × 4 matrices T4Xe and V4Xo are assigned in several subunits. The subunits and their outputs are shown in Table I. The proposed multistandard IDCT unit also contains an adder tree subunit. It not only accumulates the outputs of the subunits to compute the matrices T4Xe and V4Xo, but it also contains the computation of the butterfly matrix P8,l. In the adder tree subunit, addition operations are executed in tree structure for reducing the number of cycles. The outputs of the subunits are subsequently input into an adder tree subunit to compute and obtain all elements of T8X8 finally. In the design of subunit circuits, the strategies of FS (7) and AS (8) are jointly employed for reducing the circuit resources. The proposed architecture is high parallel. Eight inverse quantization data are parallel input into the IDCT architecture and then a row (column) of IDCT data is obtained in the output ports after several cycles. The proposed top-level 1-D multistandard IDCT architecture is shown in Fig. 3.

Conclusion

We have proposed a low-cost VLSI architecture of a multistandard IDCT in this brief. IDCTs of several standards are integrated in the proposed architecture. The circuits are efficiently shared and saved based on the FS and AS strategies. It can be concluded that a high decoding capability is achieved in small

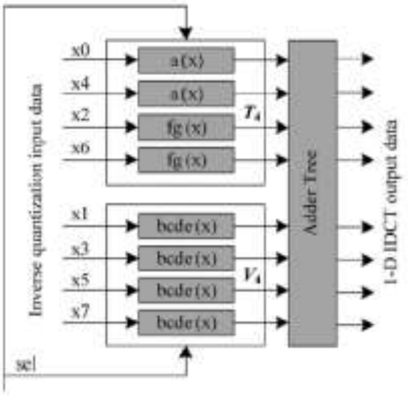
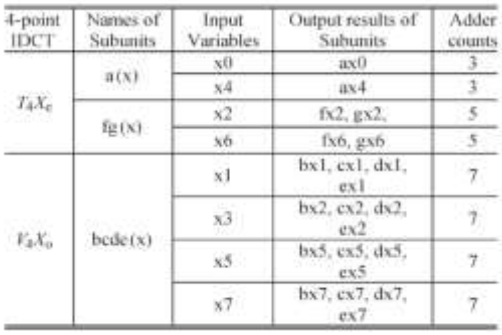


Fig. 5. Proposed top-level 1-D multistandard IDCT VLSI architecture.

TABLE III
SUBUNITS OF T_4X_e AND V_4X_o



4-point IDCT	Names of Subunits	Input Variables	Output results of Subunits	Adder counts
T_4X_e	a(x)	x0	ax0	3
		x4	ax4	3
	fg(x)	x2	fx2, gx2,	5
		x6	fx6, gx6	5
V_4X_o	bcde(x)	x1	bx1, cx1, dx1, ex1	7
		x3	bx2, cx2, dx2, ex2	7
		x5	bx5, cx5, dx5, ex5	7
		x7	bx7, cx7, dx7, ex7	7

References

[1] N. Ahmed, T. Natarajan, and K. R. Rao, “On image processing and a discrete cosine transform,” IEEE Trans. Comput., vol. C-23, no. 1, pp. 90–93, Jan. 1974.

[2] S. Lee and K. Cho, “Architecture of transform circuit for video decoder
Supporting multiple standards,” Electron. Lett., vol. 44, no. 4, pp. 274–275, Feb. 2008.

[3] N. T. Ngo, T. T. T. Do, T. M. Le, Y. S. Kadam, and A. Bermak, “ASIPcontrolled
inverse integer transform for H.264/AVC compression,” in Proc. IEEE/IFIP Int. Symp. Rapid Syst. Prototyping, 2008, pp. 158–164.

[4] Y. Li, Y. He, and S. L. Mei, “A highly parallel joint VLSI architecture for
transforms in H.264/AVC,” J. Signal Process. Syst., vol. 50, no. 1, pp. 19–32, Jan. 2008.

[5] C. P. Fan and G. A. Su, “Fast algorithm and low-cost hardware-sharing design of multiple integer transforms for VC-1,” IEEE Trans. Circuits Syst.II, Exp. Briefs, vol. 56, no. 10, pp. 788–792, Oct. 2009.

[6] W. Chen, C. H. Smith, and S. C. Fralick, “A fast computational algorithm for the discrete cosine transform,” IEEE Trans. Commun., vol. COM-25, no. 9, pp. 1004–1009, Sep. 1977.

[7] IEEE Standard Specifications for the Implementations of 8 × 8 Inverse Discrete Cosine Transform, IEEE Standard 1180-1990, 1990.

[8] T. C. Wang, Y. W. Huang, H. C. Fang, and L. G. Chen, “Parallel 4 × 4 2D transform and inverse transform architecture for MPEG-4 AVC/H.264,” in Proc. IEEE Int. Symp. Circuits Syst., 2003, pp. 800–803.