A Low-Cost Very Large Scale Integration Architecture for Multistandard Inverse Transform

Honggang Qi, Qingming Huang, *Senior Member, IEEE*, and Wen Gao, *Fellow, IEEE*

***Abstract*—In this brief, a low-cost very large scale integration (VLSI) 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 circuit share strategies, factor share (FS) and adder share (AS), are applied to the inverse transform architecture for saving its circuit resource. It is shown that the proposed multistandard inverse transform architecture can support the real-time decoding of 1920 $\times$ 1080@60 Hz high-definition video at the cost of low circuit resource.**

***Index Terms*—Circuit share, high-definition video, multistandard inverse transform, very large scale integration (VLSI).**

I. Introduction

CURRENTLY, several video compression standards, e.g., MPEG-2, MPEG-4 ASP, H.264/AVC and VC-1 (Windows Media Video 9), are widely applied in video codec products, such as digital TV, mobile video, video conference, and so on. The intercommunications between the video devices using different standards are so much inconvenient, thus video codec supporting multiple standards are more useful and more attractive. Integrating multistandard encoding or decoding circuits into a single chip will increase the area and power consumed, which has a negative impact on the cost of the chip products. Thus, a critical problem that is needed to be solved in multistandard codecs is the increasing cost. It will be competitive that multistandard codecs achieve both high performance and low cost. The circuit share is an efficient method for the circuit resource reduction. The integration of multistandard codec does not mean that several standard codecs 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.

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. Similarly, in this brief, a low-cost VLSI architecture of a multistandard IDCT is also proposed for the real-time decoding of HD1080P@60 Hz video.

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.

II. 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 II-type IDCT T_8 can be expressed in matrix form as follows:

$$T_8 = \begin{bmatrix} a & b & f & c & a & d & g & e \\ a & c & g & -e & -a & -b & -f & -d \\ a & d & -g & -b & -a & e & f & c \\ a & e & -f & -d & a & c & -g & -b \\ a & -e & -f & d & a & -c & -g & b \\ a & -d & -g & b & -a & -e & f & -c \\ a & -c & g & e & -a & b & -f & d \\ a & -b & f & -c & a & -d & g & -e \end{bmatrix} \quad (1)$$

where $a \sim g$ stand for seven different transform elements in the IDCT matrix. The implementation of the IDCT needs a number of multiplication and addition operations according to its definition. For example, the 8-point IDCT of a row or column vector needs 64 multiplications and 56 additions. Obviously,

Manuscript received December 21, 2009; revised February 1, 2010; accepted March 15, 2010. Date of publication June 1, 2010; date of current version July 16, 2010. This work was supported in part by the Natural Science Foundation of China under Grant 60833006 and in part by the National Basic Research Program of China under Grant 2009CB320905. This paper was recommended by Associate Editor Z. Wang.

The authors are with the College of Information Sciences and Engineering, Graduate University of the Chinese Academy of Sciences, Beijing 100049, China (e-mail: hgqi@jdl.ac.cn; qmhuang@jdl.ac.cn; wgao@jdl.ac.cn).

Digital Object Identifier 10.1109/TCSII.2010.2048477

the computation complexity is so high that it is difficult to realize IDCT directly in both software and hardware environments. It has been known that there are redundant operations in the IDCT. The redundant operations may be reduced using the properties of IDCT such as the recursion property. 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 T_8 can be decomposed into a 4-point II-type IDCT and a 4-point IV-type IDCT. The decomposition of a 1-D 8-point IDCT is expressed as follows:

$$T_8 = P_{8,l}\begin{bmatrix} T_4 & 0 \\ 0 & V_4 \end{bmatrix} P_{8,r} \tag{2}$$

where T_4 is the 4-point II-type IDCT and V_4 is the 4-point IV-type IDCT.

$$T_4 = \begin{bmatrix} a & f & a & g \\ a & g & -a & -f \\ a & -g & -a & f \\ a & -f & a & -g \end{bmatrix} \quad V_4 = \begin{bmatrix} e & d & c & -b \\ d & b & e & c \\ c & e & -b & -d \\ b & -c & d & e \end{bmatrix}.$$

$P_{8,l}$ and $P_{8,r}$ are two 8×8 matrices only including elements of 1 and 0 [6]. $P_{8,l}$ may be implemented in butterfly structure with 8 additions. $P_{8,r}$ is the permutation matrix without any arithmetic operation. The transform of a 1-D IDCT is expressed as follows:

$$\begin{aligned} T_8X_8 &= P_{8,l}\begin{bmatrix} T_4 & 0 \\ 0 & V_4 \end{bmatrix} P_{8,r}X \\ &= P_{8,l}\begin{bmatrix} T_4 & 0 \\ 0 & V_4 \end{bmatrix}\begin{bmatrix} X_{8e} \\ X_{8o} \end{bmatrix} = P_{8,l}\begin{bmatrix} T_4X_{8e} & 0 \\ 0 & V_4X_{8o} \end{bmatrix} \end{aligned} \tag{3}$$

where

$$\begin{aligned} X_8 &= [x0 \quad x1 \quad x2 \quad x3 \quad x4 \quad x5 \quad x6 \quad x7]^T \\ X_{8e} &= [x0 \quad x2 \quad x4 \quad x6]^T \\ X_{8o} &= [x1 \quad x3 \quad x5 \quad x7]^T \\ T_4X_{8e} &= \begin{bmatrix} a & f & a & g \\ a & g & -a & -f \\ a & -g & -a & f \\ a & -f & a & -g \end{bmatrix}\begin{bmatrix} x0 \\ x2 \\ x4 \\ x6 \end{bmatrix} \\ &= \begin{bmatrix} ax0 + fx2 + ax4 + gx6 \\ ax0 + gx2 - ax4 - fx6 \\ ax0 - gx2 - ax4 + fx6 \\ ax0 - fx2 + ax4 - gx6 \end{bmatrix} \\ V_4X_{8o} &= \begin{bmatrix} e & d & c & -b \\ d & b & e & c \\ c & e & -b & -d \\ b & -c & d & e \end{bmatrix}\begin{bmatrix} x1 \\ x3 \\ x5 \\ x7 \end{bmatrix} \\ &= \begin{bmatrix} ex1 + dx3 + cx5 - bx7 \\ dx1 + bx3 + ex5 + cx7 \\ cx1 + ex3 - bx5 - dx7 \\ bx1 - cx3 + dx5 + ex7 \end{bmatrix}. \end{aligned}$$

The implementation of T_4 or V_4 needs 16 multiplications and 12 additions. The implementation of butterfly matrix $P_{8,l}$ needs 8 additions. In total, the decomposition implementation of a 1-D 8-point IDCT needs 32 (16×2) multiplications and 32 ($12 \times 2 + 8$) additions. The 4-point II-type IDCT T_4 can be further decomposed into two 2-point IDCTs according to the recursion property. Through the repeatedly recursive decomposition, the redundant operations are greatly removed.

The elements of the IDCT matrix $a \sim f$ 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 codecs 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 $a \sim f$ 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 follows:

$$T_{4,\mathrm{AVC}} = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix}$$

$$T_{4,VC1} = \begin{bmatrix} 17 & 22 & 17 & 10 \\ 17 & 10 & -17 & -22 \\ 17 & -10 & -17 & 22 \\ 17 & -22 & 17 & -10 \end{bmatrix}$$

$$T_{8,\mathrm{AVC}} = \begin{bmatrix} 8 & 12 & 8 & 10 & 8 & 6 & 4 & 3 \\ 8 & 10 & 4 & -3 & -8 & -12 & -8 & -6 \\ 8 & 6 & -4 & -12 & -8 & 3 & 8 & 10 \\ 8 & 3 & -8 & -6 & 8 & 10 & -4 & -12 \\ 8 & -3 & -8 & 6 & 8 & -10 & -4 & 12 \\ 8 & -6 & -4 & 12 & -8 & -3 & 8 & -10 \\ 8 & -10 & 4 & 3 & -8 & 12 & -8 & 6 \\ 8 & -12 & 8 & -10 & 8 & -6 & 4 & -3 \end{bmatrix}$$

$$T_{8,\mathrm{VC1}} = \begin{bmatrix} 12 & 16 & 16 & 15 & 12 & 9 & 6 & 4 \\ 12 & 15 & 6 & -4 & -12 & -16 & -16 & -9 \\ 12 & 9 & -6 & -16 & -12 & 4 & 16 & 15 \\ 12 & 4 & -16 & -9 & 12 & 15 & -6 & -16 \\ 12 & -4 & -16 & 9 & 12 & -15 & -6 & 16 \\ 12 & -9 & -6 & 16 & -12 & -4 & 16 & -15 \\ 12 & -15 & 6 & 4 & -12 & 16 & -16 & 9 \\ 12 & -16 & 16 & -15 & 12 & -9 & 6 & -4 \end{bmatrix}. \tag{4}$$

MPEG-2 and MPEG-4 ASP standards only define the real IDCT, instead of the integer IDCT, but they permit user-defined integer IDCTs and the user-defined integer IDCTs are required to conform to the constraint of IEEE 1180-1990 [7]. In this brief, the integer transform of MPEG-2/4 adopts a 10-bit integer transform, which is the transform meeting the constraint of IEEE 1180-1990 with the minimum bitwidth (5), as shown at the bottom of the next page. In addition, the 8-point and 4-point Hadamard transforms are used in H.264/AVC. They are also integrated into the proposed multistandard IDCT architecture.

III. 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. For example, element 473 can be factorized as $2^8 - 2^5 - 2^3 + 2^0$. The expression can be friendly implemented with additions and shifts in the digital system. 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. FS helps to avoid the repeating circuit implementations of these same SBFs. Let E_0 and E_1 denote two different elements of the IDCT matrix, the binary factorization of E_0 and E_1 are expressed as

$$E_0 = \sum_{i=0}^{N} r_i 2^i, \quad E_1 = \sum_{i=0}^{N} s_i 2^i, \quad r_i, s_i = \pm 1, 0, \quad N \geq 0 \tag{6}$$

where N is the maximum bitwidth of all integer elements. Extracting the same SBFs $e_{\Delta m}$ from E_0 and E_1, then (6) is rewritten as

$$\begin{aligned} E_0 &= \sum_{m=0}^{M-1} 2^{j_m} e_{\Delta m} + e_0 \\ E_1 &= \sum_{m=0}^{M-1} 2^{k_m} e_{\Delta m} + e_1 \\ e_{\Delta m} &= \sum_{i=0}^{N} t_i 2^i, \quad t_i = \begin{cases} 0, & r_{i+j_m} \neq s_{i+k_m} \\ r_{i+j_m}, & r_{i+j_m} = s_{i+k_m} \end{cases} \\ e_0 &= \sum_{i=0}^{N} u_i 2^i, \quad u_i = \begin{cases} r_i, & r_i \neq s_i \\ 0, & r_i = s_i \end{cases} \\ e_1 &= \sum_{i=0}^{N} v_i 2^i, \quad v_i = \begin{cases} s_i, & r_i \neq s_i \\ 0, & r_i = s_i \end{cases} \end{aligned} \tag{7}$$

where $\sum_{m=0}^{M-1} 2^{j_m} e_{\Delta m}$ and $\sum_{m=0}^{M-1} 2^{k_m} e_{\Delta m}$ are the sums of the SBFs, and e_0 and e_1 are the remainder SBFs of elements E_0 and E_1, excluding $\sum_{m=0}^{M-1} 2^{j_m} e_{\Delta m}$ and $\sum_{m=0}^{M-1} 2^{k_m} e_{\Delta m}$. Applying FS to the circuit implementation of IDCT, $e_{\Delta m}$, e_0, and e_1 are first computed at the same time, and then these intermediate results are used for computing E_0 and E_1. It is not necessary to implement two circuits of the same SBFs $e_{\Delta m}$ for computing E_0 and E_1, respectively. One circuit of the same SBFs $e_{\Delta m}$ can be shared in the computations of E_0 and E_1. $2^{j_m} e_{\Delta m}$ can be implemented by j_m-bit left-shift

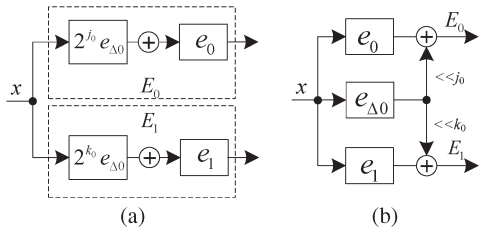

Fig. 1. Circuit of element factorization (a) without FS; (b) with FS.

TABLE I
BINARY FACTORIZATIONS OF MPEG-2/4 IDCT BASED ON FS

Symbols	Elements	Factorizations
a	362	[〖4+2〗 <<6]-16- 〖4+2〗
f	473	[〖128+64+4〗 <<1]+64+16+1
g	196	〖128+64+4〗
b	502	[256 - 〖4+1〗]<<1
c	426	[128+[〖4+1〗 <<4]+ 〖4+1〗]<<1
d	284	[128+ 〖16-2〗]<<1
e	100	[64- 〖16-2〗]<<1

〖 〗 : shared factor, << : left-shift operator.

of $e_{\Delta m}$. The circuit comparison of applying and not applying FS in the condition of $M = 1$ is presented in Fig. 1. Equation (7) is a basic mathematical analysis. The binary factorization of an element is not unique. Different binary factorization forms have different share factors e_Δ. However, it is fortunate that the elements of the IDCT are usually not complicated. It is possible to obtain an optimal or near-optimal e_Δ reducing the circuits as possible. The elements of MPEG-2/4 IDCT matrix is the most complicated in several IDCTs to be integrated in the proposed multistandard IDCT architecture. The elaborate binary factorizations of the elements of MPEG-2/4 IDCT based on FS are shown in Table I. After 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.

In the multistandard IDCT architecture, only an IDCT is enabled when decoding a standard bitstream. The circuits from other disabled IDCTs are in sleep mode. If these sleep circuits can be made full use of, more circuits will be saved. In the multiplier-less implementation of IDCT, the adder consumes more circuit resources than other operators; thus, AS is employed to share the adders among different IDCTs. If M IDCTs

$$T_{8,\mathrm{MP24}} = \begin{bmatrix} 362 & 502 & 473 & 426 & 362 & 284 & 196 & 100 \\ 362 & 426 & 196 & -100 & -362 & -502 & -473 & -284 \\ 362 & 284 & -196 & -502 & -362 & 100 & 473 & 426 \\ 362 & 100 & -473 & -284 & 362 & 426 & -196 & -502 \\ 362 & -100 & -473 & 284 & 362 & -426 & -196 & 502 \\ 362 & -284 & -196 & 502 & -362 & -100 & 473 & -426 \\ 362 & -426 & 196 & 100 & -362 & 502 & -473 & 284 \\ 362 & -502 & 473 & -426 & 362 & -284 & 196 & -100 \end{bmatrix} \tag{5}$$

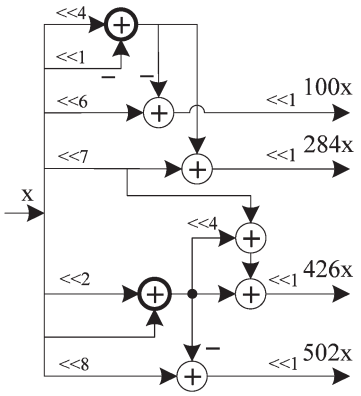

Fig. 2. Proposed circuit architecture of subunit $bcde(x)$ of MPEG-2/4 8-point IDCT based on FS.

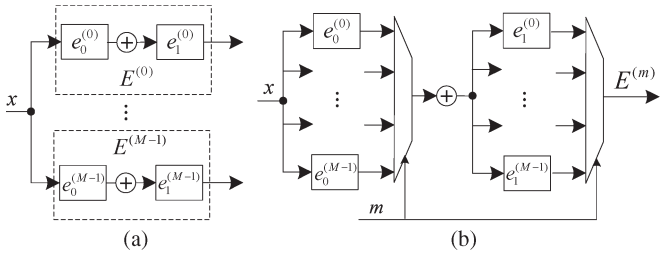

Fig. 3. Circuit of element factorization (a) without AS, (b) with AS.

TABLE II
BINARY FACTORIZATIONS OF MPEG AND VC-1 BASED ON AS

Symbols	MPEG-2/4		VC-1 (8-point)	
	Elements	Factorizations	Elements	Factorizations
a	362	[(4⊕2)<<6]-16-(4⊕2)	12	(8⊕4)
f	473	[〖128+64+4〗<<1]+64+16⊕1	16	16
g	196	〖128+64+4〗	6	(4⊕2)
b	502	[256-(4+1)]<<1	16	16
c	426	[128+[(4⊕1)<<4]+(4⊕1)]<<1	15	(16⊕-1)
d	284	[128+(16⊕-2)]<<1	9	(8⊕1)
e	100	[64-(16⊕-2)]<<1	4	4

〖 〗: shared factor, ⊕ : shared adder, << : left-shift operator.

are integrated in a multistandard IDCT architecture, then the AS strategy is expressed as

$$e^{(m)} = \left\langle \begin{matrix} e_0^{(0)} \\ \vdots \\ e_0^{(M-1)} \end{matrix} \right\rangle_m + \left\langle \begin{matrix} e_1^{(0)} \\ \vdots \\ e_1^{(M-1)} \end{matrix} \right\rangle_m, \quad m \in \{0, \ldots M-1\} \tag{8}$$

where $e_0^{(m)}$ and $e_1^{(m)}$ are two factorized factors of an element of the mth IDCT matrix, $e^{(m)}$ is the sum of $e_0^{(m)}$ and $e_1^{(m)}$, i.e., $e^{(m)} = e_0^{(m)} + e_1^{(m)}$, $\langle\ \rangle$ is the multiplexer operator, and $\langle\ \rangle_m$ denotes that the mth element is selected from all vector elements. From the AS strategy (8), it is derived that an adder can be shared to compute the sums of factors by different IDCTs. The circuit comparison of applying and not applying AS is presented in Fig. 3. In Fig. 3(b), some multiplexers are added in the circuit for selecting the factors of the enabled IDCT. The added multiplexers cannot increase the circuit cost significantly because the multiplexer needs very few gates than the adder.

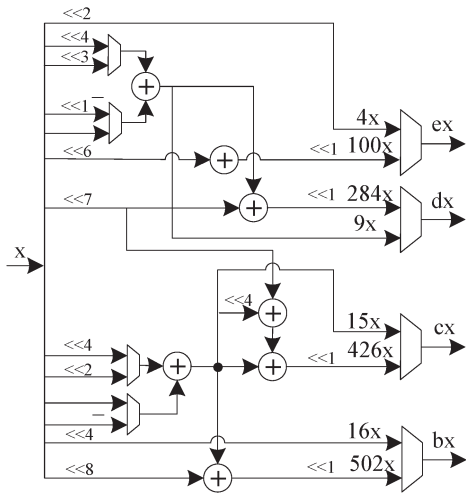

Fig. 4. Proposed circuit architecture of the subunit $bcde(x)$ of integrating MPEG-2/4 and VC-1 8-point IDCTs based on AS.

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

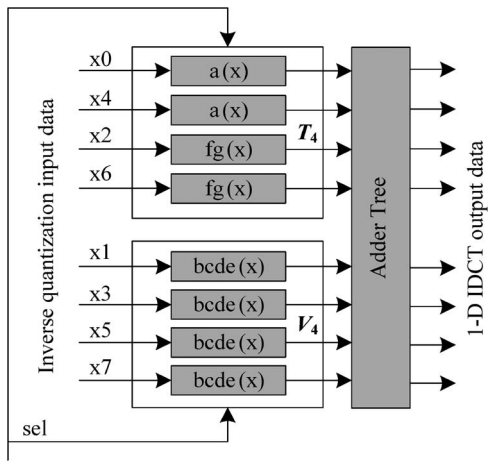

Fig. 5. Proposed top-level 1-D multistandard IDCT VLSI architecture.

Theoretically, applying AS to the multistandard IDCT architecture composed of M IDCTs, M-1 adders can be saved. As an example, the binary factorizations of MPEG-2/4 and VC-1 8-point IDCTs based on AS are shown in Table II. It is seen from Table II that, compared with the circuit of a single MPEG-2/4 IDCT, no adder is increased in the circuit of

TABLE IV
COMPARISONS BETWEEN EXISTING AND PROPOSED 1-D IDCT ARCHITECTURES

Architecture	Gate counts	Decoding capability	Working frequency	Technology	Supporting Standards		
					MPEG-2/4	H.264/AVC	VC1
Lee's[2]	19.1 K	1920x1080@22Hz	136 MHz	0.18	Y	Y	Y
Ngo's[3]	21.5K	4096x2304@30Hz	176 MHz	0.35	N	Y	N
Li's[4]	13.7K	1920x1080@30Hz	200 MHz	0.18	N	Y	N
Fan's[5]	9.9K	No report	100 MHz	0.18	N	N	Y
Proposed	18 K	1920x1080@60Hz	100 MHz	0.13	Y	Y	Y

integrating two IDCTs. The proposed circuit architecture of the subunit $bcde(x)$ is shown in Fig. 4.

IV. 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 T_4 and V_4, a butterfly matrix $P_{8,l}$, and a permutation matrix $P_{8,r}$. The computations of 1×4 matrices T_4X_e and V_4X_o are assigned in several subunits. The subunits and their outputs are shown in Table III. The proposed multistandard IDCT unit also contains an adder tree subunit. It not only accumulates the outputs of the subunits to compute the matrices T_4X_e and V_4X_o, but it also contains the computation of the butterfly matrix $P_{8,l}$. 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 T_8X_8 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 parallelly 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. 5.

In the proposed architecture, the 4-point transform T_4 is implemented with 22 adders and the 4-point transform V_4 is implemented with 40 adders. Including 8 adders of the butterfly addition operations outside of T_4 and V_4 transforms, the 1-D IDCT architecture needs 70 $(22 + 40 + 8)$ adders in total. The throughputs of the architecture are 8 pixels per cycle in an 8-point inverse transform and 4 pixels per cycle in a 4-point inverse transform. To reduce the delay of critical paths, 4-stage buffers are added in the architecture. Thus, the latency of the pipeline architecture is 4 cycles. It takes only 11 $(4 + 7)$ cycles to finish the process of a horizontal (vertical) 1-D 8×8 IDCT in the pipeline way. It is known that a two-dimensional (2-D) IDCT can be implemented with two separate 1-D IDCTs. The proposed 1-D IDCT architecture is implemented twice in decoder to fulfill 2-D IDCT of the video data. If a Chen's transpose RAM [8] is employed to buffer and transpose the intermediate data between the horizontal and vertical 1-D IDCTs, it takes 103 cycles to process a 2-D 8×8 IDCT of a 4:2:0 MB in the pipeline way. The proposed VLSI architecture of the multistandard IDCT can decode 1080P@60 Hz 4:2:0 HD video bitstream real-time in 50 MHz $(1920 \times 1088 \times 60 \times 103/256)$ working frequency. It as an IP can be embedded into real-time HD multistandard decoder to decode MPEG-2/4, H.264/AVC and VC-1 video bitstreams.

V. COMPARISONS OF SYNTHESIZED RESULTS

The proposed 1-D multistandard IDCT architecture excluding the transpose RAM is simulated in Virtex-II Pro FPGA and synthesized with TSMC 0.13-μm CMOS standard cell technology. The synthesized results show that the proposed architecture costs about 18k logic gates in the maximum working frequency 100 MHz. The working frequency can be further improved to 200 MHz by inserting the registers into some critical paths at the cost of a very small circuit area increase. Table IV shows the comparisons of the synthesis results of several similar 1-D IDCT architectures [2]–[5]. It is seen from the comparison that the proposed 1-D IDCT architecture has a fewer gate count and higher decoding capability than Lee's [2] and supports more IDCTs than Ngo's [3], Li's [4], and Fan's [5].

VI. 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 IDCT architecture area.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers. Their valuable comments are appreciated.

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, "ASIP-controlled 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.