CODING SENSITIVE BASED APPROXIMATION ALGORITHM FOR POWER EFFICIENT VBS-DCT VLSI DESIGN IN HEVC HARDWIRED INTRA ENCODER

Liangliang Chang†, Zhenyu Liu⋆, Xiangyang Ji‡, Dongsheng Wang⋆

⋆RIIT&TNList/‡Department of Automation/†IMETU, Tsinghua University, Beijing 100084, China. E-mail: liuzhenyu73@tsinghua.edu.cn

ABSTRACT

High Efficiency Video Coding (HEVC), emerging as the latest video coding standard, obtained a 50% bit-rate reduction while maintaining the competitive visual quality as H.264/AVC. Rate-Distortion Optimization (RDO) is a computation intensive module in HEVC encoding. In specific, during Intra coding, RDO accounts for 62% of the overall encoding time. The 2-dimensional DCT is the most area and power consuming component for VLSI implementation of RDO module. In this paper, we decompose the matrix multiplication of DCT into several sparse butterfly structures in series. In addition, the computation and the storage of 25% high frequency coefficients are dropped by our approximation algorithm. The proposed algorithms are integrated in HM15.0. It is verified that our methods could save 15.9% time with 1.03% BDBR augment. We further implement the DCT VLSI design using TSMC 90nm standard cell library. In worst conditions (125°C, 0.9V), the power dissipation of our DCT is 12.7mW at the 311MHz maximum clock speed. As compared to the primitive design, we achieved 71.9% of hardware and 70.2% of power reductions.

***Index Terms*—** DCT, Intra Prediction, HEVC, Sparse Butterfly Decomposition, VLSI

1. INTRODUCTION

The High Efficiency Video Coding (HEVC) standard is the up-to-date video coding standard jointly developed by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG)[1]. As compared to H.264/AVC, HEVC could save 50% rates with the same perceptual quality, while increasing the encoder complexity by 5x-10x [2]. Therefore, it is challenging to design the power-efficient HEVC encoder that maintains the high compression performance. Especially, HEVC devises a flexible partitioning strategy with the concepts of coding unit (CU), prediction unit (PU), and transform unit (TU) that enhance the coding efficiency. The recursive partition processes and the various configurations of CU/PU/TU modes seriously increase the complexity of Rate-Distortion Optimization (RDO). In the RDO based modes decision process, nearly 60% computation cost stems from the discrete cosine transform (DCT) processing. Accordingly, in hardwired HEVC encoder design, more than 20% of the overall power is consumed by DCT module.

To meet the low-power real-time coding requirements, numerous fast DCT algorithms are proposed. There are two classical methods proposed to reduce DCT computation complexity[3][6]. In literature [3], Hein decomposed a $N \times N$ DCT matrix as the product of one $N \times N$ Hadamard matrix and a $N \times N$ block diagonal matrix. Ahmed, et al, [4] realizes the DCT engines adopted in HEVC RDO by using the method of literature [3]. In [5], Masera tested the sensitivity of the coefficients in the block diagonal matrix to the coding quality, and then skip the multiplications of trivial coefficients. The main drawback of [3] is that the hardware can not be shared by the variable-block-size (VBS) DCT. To overcome this hindrance, Chen, et al., proposed the recursive decomposition DCT architecture [6]. The DCT employed by HEVC also stems from [6]. The main difference between HEVC and [6] is that the $\mathbf{R}$-matrix in HEVC is not further decomposed as the literature [6], because this can achieve the higher transformation precision with the limited 16-bit word width. Zhu provided the VLSI VBS-DCT engine [7], which illustrated the advantages in terms of hardware sharing and high clock speed as compared to [4]. To further reduce the hardware costs, Zhu replaced the 8/16/32 DCTs with 8/16/32 Hadamard Transforms, respectively, which introduced the averaged +1.61% BDBR and 15.9% time saving. [8] simplifies the big block ($N > 4$) in [6] with 4×4 transforms. However, the coding quality loss is serious. For example, at the high bit-rate, the quality loss for BasketballDrill was about 2dB.

In this paper, according to the coding quality sensitive analysis, we approximate the DCT by decomposing the $\mathbf{R}$-matrix into several sparse butterfly-structure multiplications in series as [6] , and further eliminate 25% computation in the row- and column-wise 1D transforms. The proposed algorithms outperform the counterpart [7] in terms of coding quality, hardware-cost and power-cost saving.

The rest of this paper are organized as follows. In section 2, we briefly introduce the simplified RDO algorithm for HEVC intra coding, which employs the proposed low-cost hardwired DCT engine. In section 3, we describe our DCT matrix decomposition method and the associated VLSI design. Experimental results are illustrated in section 4, followed by the conclusions in section 5.

2. SIMPLIFIED RDO IN HEVC INTRA CODING

HEVC standard proposes the concept of quadtree structure coding unit (CU). For one $2N \times 2N$ CU, it can be coded as a whole ($2N \times 2N$ mode), or be split into four $N \times N$ sub-CUs. The root node of a CU quadtree is denoted as coding tree unit (CTU). Each leaf node of a CU quadtree contains its prediction unit (PU) quadtree and its transform unit (TU) quadtree. PU indicates the prediction information of each CU, such as the prediction unit size and prediction methods. TU quadtree of a CU determines its transform unit structure. In one CU's encoding procedure, RDO is carried out to derive the most efficient PU and TU structures. Let M_{PU} and M_{TU} denote the CU/TU configurations, respectively. The HEVC encoder traverses various combinations $\vec{M} = (M_{PU}, M_{TU})$ to find the optimal candidate with the minimum rate-distortion cost (RD-cost), expressed as

$$\vec{M}_o = \arg\min_{\vec{M}} \{D(\vec{M}) + \lambda \cdot R(\vec{M})\},$$

This work is supported by Projects of International Cooperation and Exchanges NSFC 61325003 and National Key R&D Plan (No.2016YFB0200505).

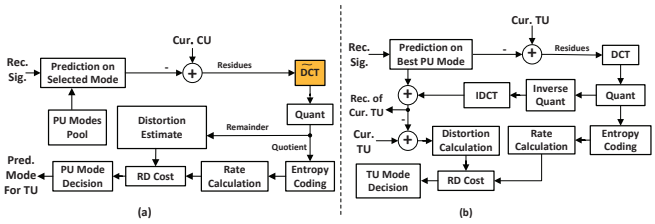

Fig. 1. Simplified RDO Procedure for Optimal PU and TU Mode Decision ((a) PU mode decision procedure with low-cost DCT (b) TU mode decision procedure)

in which, $D(\vec{M})$ and $R(\vec{M})$ represent the distortion and rate overheads of mode $\vec{M}$, respectively, and λ is the Lagrangian multiplier.

In HEVC reference software, the PU and TU mode are determined in sequential order. That is, for a CU, we first define the best PU mode, and then determine the best TU mode for the residues. Zhu proposed a low complex RDO for PU mode decision [9], where the pseudo DCT ($\widetilde{\mathrm{DCT}}$) substitutes the original DCT as shown in Fig. 1(a). During TU mode RDO, the original DCT must be adopted to avoid drifting problem, as shown by Fig. 1(b).

3. CODING SENSITIVE BASED APPROXIMATED DCT

In order to simplify the arithmetics in DCT, which consequently saves the hardware cost and power consumption, we propose a approximation algorithm to decompose the DCT matrix and devise the corresponding hardware architecture in this section. In subsection 3.1, we describe the method to decompose the DCT matrix into butterfly structures. Subsection 3.2 discusses our sensitive base high frequency coefficient approximation scheme. In 2D DCT engine, the transpose register file (TRF) accounts for more than 30% chip area. Fortunately, our approximation drops the high frequency storage, which contributes to the optimization of TRF. In subsection 3.3, we propose the transpose memory design.

3.1. DCT Matrix Decomposition

Because DCT could eliminate the correlations among prediction residues and concentrate the residues' energy [10], it is applied in various image and video compression standards. HEVC standard adopts the 16-bit precision integer DCT. That is, the coefficients in DCT matrix is scaled and rounded into 16-bit integer numbers, which take both of the precision and the closeness to orthogonality into consideration[2]. For example, the 4×4 integer DCT transform matrix is

$$\mathbf{A}_4 = \begin{bmatrix} 64 & 64 & 64 & 64 \\ 83 & 36 & -36 & -83 \\ 64 & -64 & -64 & 64 \\ 36 & -83 & 83 & -36 \end{bmatrix} \quad (1)$$

The VBS-DCT employed in HEVC reference software(HM) [11] is generated from the literature [6], in which the original DCT matrix is decomposed as the multiplication of a butterfly structure and a block diagonal matrix. In specific, the $N \times N$ DCT matrix is formulated as

$$\mathbf{A}_N = \mathbf{P}_N \begin{bmatrix} \mathbf{A}_{\frac{N}{2}} & \mathbf{O}_{\frac{N}{2}} \\ \mathbf{O}_{\frac{N}{2}} & \mathbf{R}_{\frac{N}{2}} \end{bmatrix} \mathbf{B}_N, \quad (2)$$

where $\mathbf{P}_N$ is a permutation matrix that changes the output data from its natural order to bit-reversed order. $\mathbf{B}_N$ represents the butterfly

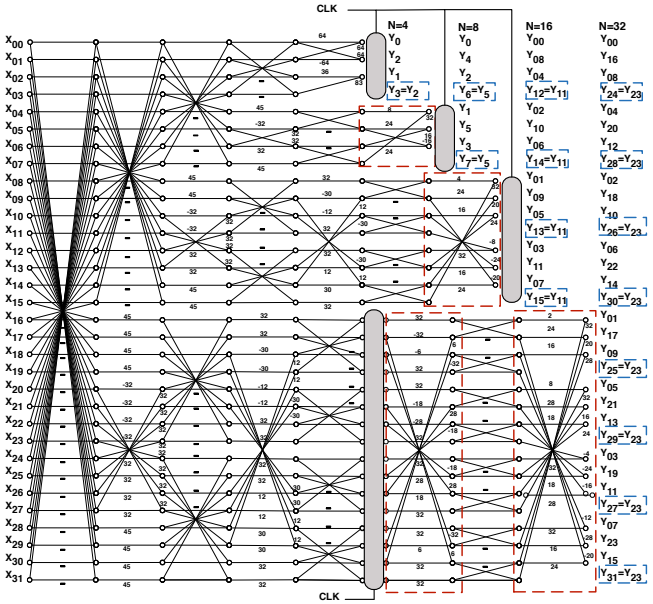

Fig. 2. The whole structure of our proposed VLSI architecture for variable block size approximate DCT

structure. $\mathbf{R}_{\frac{N}{2}}$ constitutes the information of even rows in $\mathbf{A}_N$. For example, $\mathbf{R}_2$ is used to build the 2nd and the 4th rows of (1) and could be expressed as

$$\mathbf{R}_2 = \begin{bmatrix} 83 & 36 \\ 36 & -83 \end{bmatrix}$$

Equation (2) is a recursive decomposition. The initial one $\mathbf{A}_2$ is

$$\mathbf{A}_2 = \begin{bmatrix} 64 & 64 \\ 64 & -64 \end{bmatrix}$$

The circulant matrix like $\mathbf{R}_{N/2}$ could be further decomposed into several stages of butterfly structure multiplication[12]. In literature [6], the matrix $\mathbf{R}_{N/2}$ is decomposed as

$$\mathbf{R}_{\frac{N}{2}} = \mathbf{M}1 \cdot \mathbf{M}2 \cdot \mathbf{M}3 \cdot \mathbf{M}4 \cdot \ldots \cdot \mathbf{M}(2log_2N - 3) \quad (3)$$

The detailed definitions of matrices $\mathbf{M}n$ are provided in [6]. Each matrix $\mathbf{M}n$ has a butterfly structure. When n is odd, the elements of matrix $\mathbf{M}n$ are floating-point numbers. However, HEVC standards did not decompose $\mathbf{R}_{N/2}$, because rounding those floating-point matrices $\mathbf{M}n$ will introduce much noises with the 16-bit integer expression.

On the other hand, the method of (3) can reduce the arithmetics efficiently. On the basis of (3), we approximate DCT in HEVC PU modes decision stage. That is the approximated $\mathbf{R}_{N/2}$, i.e., $\widetilde{\mathbf{R}}_{N/2}$ has the form of

$$\widetilde{\mathbf{R}}_{\frac{N}{2}} = [45\mathbf{M}1] \cdot \mathbf{M}2 \cdot [32\mathbf{M}3] \cdot \mathbf{M}4 \cdot \ldots \cdot [32\mathbf{M}(2log_2N - 3)], \quad (4)$$

where the operation $[\cdot]$ is rounding. The proposed VLSI architecture for VBS approximated DCT is depicted in Fig.2. As the rounding errors of $\mathbf{M}1$ will be accumulated in all following stages [13], we assign the largest scale factor 45 to $\mathbf{M}1$. The rest stages are scaled by 32 to guarantee the minimal coefficient amplitude after rounding not less than 1. We further simplified the multipliers in the last few stages, which are indicated by the red dash line blocks in Fig.2. For

Table 1. Coefficients of original and simplified $[32\mathbf{M}5]$ in $\widetilde{\mathbf{R}}_8$

Original	28	9	31	15	25	3
Simplified	24	8	32	16	24	2

Table 2. Definition of $s_{N/2}$

$\widetilde{\mathbf{R}}_{N/2}$	$\widetilde{\mathbf{R}}_4$	$\widetilde{\mathbf{R}}_8$	$\widetilde{\mathbf{R}}_{16}$
$s_{N/2}$	4	9	14

Table 3. Computational complexity comparisons

Size	Op.	Algorithms				
		DCT	Hadamard[9]	Chen[6]	Proposed	
					①	②
8×8	Mult	24	-	16	7	6
	Add	28	24	26	26	24
16×16	Mult	88	-	44	25	24
	Add	100	64	74	74	70
32×32	Mult	344	-	116	74	72
	Add	372	160	194	194	186

* ① indicates the computation cost of our DCT decomposition algorithm in Section 3.1 while ② indicates the computation cost after high frequency coefficients elimination in Section 3.2

example, the original coefficients of $[32\mathbf{M}5]$ in $\widetilde{\mathbf{R}}_8$ and the simplified counterparts are illustrated in Tab.1. As we know, the multiplication of a variable with a constant can be decomposed to addition operations. With our simplified coefficients, 6 additions are save in $[32\mathbf{M}5]$ in $\widetilde{\mathbf{R}}_8$. It should be noted that, this method is only used in the last stages to minimize errors like the principle of our scaling schemes. To keep the uniform amplitude after the pseudo DCT, additional right shift operations are required. Specifically, the outputs of $\widetilde{\mathbf{R}}_{N/2}$ will be signed right shifted by $s_{N/2}$ bits. The values of $s_{N/2}$ are provided in Tab.2.

Because there are seven butterfly stages after the decomposition of $\mathbf{R}_{16}$, and multiplications exist in every odd stage, the critical path in VLSI design increases seriously. In consequence, we employ the 2-stage pipeline structure to improve the maximum clock speed of our pseudo DCT hardwired engine, as shown in Fig.2. For example, the first four butterfly stages of $\widetilde{\mathbf{R}}_{16}$ are allocated in the first pipeline stages, while the others are arranged in the following pipeline. Under the working worst conditions (125°C, 0.9V), the pipeline technique can enhance the maximum clock frequency by 81MHz. The computational overhead comparisons between our methods and other counterparts, including the original DCT in HEVC and the literatures [6, 9], are presented in Tab.3. Compared to DCT in HEVC, the numbers of multiplications and additions in our design are reduced to 23% and 58%, respectively.

3.2. High Frequency Coefficient Approximation

Figure 3 presents the averaged amplitudes of the row-wise DCT coefficients and the final 2D DCT coefficients. It is obviously illustrated that the function of DCT is to concentrate the energy of prediction residues to the low frequency coefficients. For example, Fig.3 reveals that most energy after row-wise DCT is pushed to the first two columns. We divided the $N \times N$ row-wise transform coefficient matrix into 4 regions according to the column-major order. Namely, columns $[0, N/4-1]$ are in the region LL; columns $[N/4, N/2-1]$ construct the region LH; similarly, $[N/2, 3N/4-1]$ and $[N/2, 3N/4-1]$ are denoted as HL and HH, respectively. Experiments reveal that the energy of region HH in 8×8 blocks merely accounts for 0.09% of the overall transform coefficients. For 16×16 and 32×32 blocks, this ratio is even dropped to 0.05% and 0.01%,

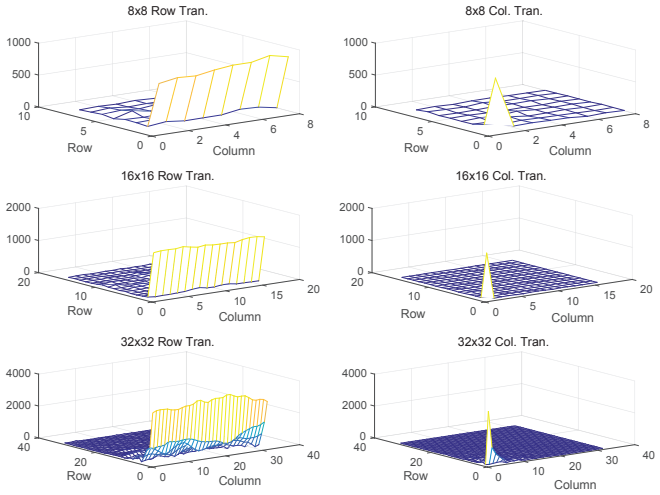

Fig. 3. Coefficient amplitude statistics of 8×8, 16×16 and 32×32 row-wise 1D DCT and final 2D DCT (The datas are extracted from the sequece: **PeopleOnStreet** ; left-hand figures: averaged coefficient amplitudes coming from the row-wise DCT; right-hand figures: averaged coefficient amplitudes of the 2D DCT)

Table 4. Coding quality comparisons after dropping 1/4 coefficients belonging to different frequency domains for 8×8, 16×16 and 32×32 DCT (BDBR: unit %)

Seq.	DCT 8×8				DCT 16×16				DCT 32×32			
	LL	LH	HL	HH	LL	LH	HL	HH	LL	LH	HL	HH
A	3.39	2.05	1.51	1.17	2.84	1.49	1.20	1.10	1.39	1.14	1.10	1.10
B	2.56	1.72	1.41	1.23	2.71	1.48	1.22	1.16	2.03	1.30	1.16	1.14
C	2.56	1.59	1.16	0.95	1.75	1.04	0.87	0.84	0.99	0.86	0.83	0.82
D	2.52	1.59	1.26	1.06	1.76	1.11	0.98	0.95	1.08	0.94	0.93	0.92
E	3.44	1.83	1.34	1.11	3.57	1.45	1.16	1.06	2.26	1.17	1.07	1.05
F	1.12	0.28	0.20	0.16	0.60	0.14	0.10	0.10	0.26	0.11	0.09	0.09
Ave.	2.60	1.51	1.14	0.95	2.20	1.12	0.92	0.87	1.33	0.92	0.86	0.85

respectively.

The coding performance sensitivity to the coefficients in the above four regions is different, which is exhibited by the experiments shown in Tab.4. For instance, dropping the coefficient in 8×8 LL introduced BDBR=2.60% compression performance degradation; In contrast, BDBR increase is merely 0.95% when dropping the coefficient in 8×8 HH. In our design, the coefficients of HH are approximated by the last column in region HL. This approximation scheme is explained in Fig.2. For example, the coefficients in 32×32 region HH, i.e., $\{Y_i | i \in [24, 32]\}$, are approximated by Y_{23}. As compared with the direct dropping method, our approximation can decrease BDBR by 0.30% in average.

3.3. Hardware Reusing Transpose Register File

The block diagram of our VBS-DCT architecture is shown as Fig.4. The VBS-DCT design is composed of the row-wise VBS transform unit, TRF, and the column-wise VBS transform unit. The aforementioned DCT approximation algorithms contribute to the chip area and power dissipation saving in VLSI design. For the simplified DCT coefficients and the predicted high frequency coefficients, the number and the complexity of arithmetic elements in row-wise and column-wise transform units are greatly reduced, which will be illustrated in section 4. In addition, because 25% coefficients after row-wise transform are derived by prediction, their storages in TRF can be dropped. In the following paragraph, we will explain the principle

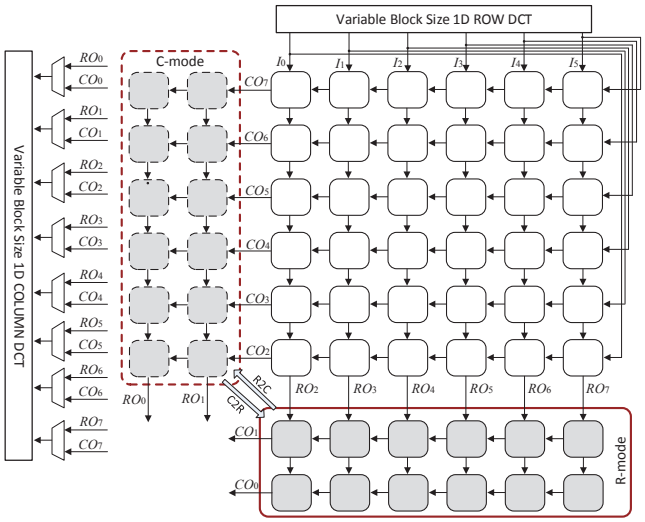

Fig. 4. Hardware reusing transpose register file architecture (Each line (I_i, RO_i and CO_i) represents 4 pixels.)

of TRF with 32×32 as an example.

In Fig.4, each square represents a basic unit that deals with 4×4 transposition. The gray squares are the reusable units. The overall capacity of TRF is 32×24 pixels. The registered data in TRF are configured to shift in horizontal or vertical, and these two configurations are interchanged with each other. The inputs of TRF come from row-wise DCT. When data are fed in from the top, in the first 24 cycles, the reusable units are configured in the C-mode and the shift direction is downward. In this period, RO_i, where $i \in [0:7]$, are dispatched to the column DCT unit. In the next 8 cycles, the reusable units are converted to R-mode that buffers the pixels from $RO_2 \sim RO_7$. In the following 32 cycles, I_i are dispatched from right that leads to the leftward shift. In the first 24 cycles of this period, the reusable units are in R-mode, and the column DCT uses CO_i as inputs; In the last 8 cycles of this period, the reusable units return to C-mode, and its function is to buffer the pixels from CO_i.

4. EXPERIMENTAL RESULTS

We realize our approximated DCT algorithm and the Hadamard based methods [9] in HM15.0 to verify their coding performance. The original HM15.0 is adopted as the anchor. The Intra coding performance analysis is presented in Tab.5.

26 typical video sequences were tested with intra_main coding configurations and QP=22,27,32,37. BDBR and BDPSNR [14] are adopted to qualitatively measure the coding efficiency of our methods. As shown in Tab.5, both algorithms obtained 15.9% time saving. The proposed method outperformed the counterpart in coding quality. In specific, our approximated DCT algorithms only cause an averaging 1.03% BDBR increase as compared to the averaged 1.61% BDBR increase from Hadamard base algorithms. Our methods possessed the superior coding performance over the Hadamard one especially for high resolution videos. For example, our BDBR of coding PeopleOnStreet is 1.17% as compared to BDBR=2.24% of Hadamard one.

We further devised the VLSI implementation of the proposed approximate DCT algorithms. The design was described with Verilog HDL and synthesized with Design Compiler using Milkway library of TSMC90nm technology. The DCT and Hadamard transform module were realized in the same way to make comparisons. As shown in Tab.6, the pipeline structure makes our design achieve the same maximum speed as the original DCT engine. Our optimizations, including the matrix decomposition, and the high frequency prediction, contribute to 82.5% hardware saving in 1D-DCT engines as compared to the primitive design. Considering the TRF optimization, the overall gate account of the proposed design is superior over Hadamard based engine [9]. In summary, our DCT engine could save 71.9% hardware cost and 70.2% power cost than the primitive design, and achieved the slightly better performance than the Hadamard counterpart. At last, but not least, $N/4$-cycle processing delay can be saved by our design.

Table 5. Coding Quality and Time Saving of Proposed RDO-based Intra Prediction Modes Decision Algorithms

Class	Sequence	Hadamard[9]			Proposed		
		BP [dB]	BR [%]	Δ [%]	BP [dB]	BR [%]	Δ [%]
A	PeopleOnStreet	-0.113	2.24	17.1	-0.060	1.17	16.2
	Traffic	-0.102	2.10	15.0	-0.053	1.09	14.2
B	BasketballDrive	-0.056	2.21	17.1	-0.040	1.56	14.9
	BQTerrace	-0.087	1.67	17.4	-0.063	1.23	14.7
	Cactus	-0.066	1.94	17.2	-0.043	1.25	16.5
	Kimono	-0.069	2.23	17.2	-0.033	1.07	15.6
	ParkScene	-0.095	2.33	16.3	-0.057	1.39	14.8
	Tennis	-0.062	2.22	15.3	-0.036	1.28	17.9
C	BasketballDrill	-0.053	1.15	14.5	-0.038	0.81	16.5
	BasketballDrillText	-0.057	1.10	14.4	-0.040	0.78	16.4
	BQMall	-0.078	1.44	14.7	-0.054	0.99	14.6
	PartyScene	-0.089	1.27	14.5	-0.065	0.92	14.3
	RaceHorsesC	-0.098	1.63	14.8	-0.067	1.11	14.4
D	BasketballPass	-0.087	1.56	15.2	-0.055	0.99	16.5
	BlowingBubbles	-0.090	1.61	16.3	-0.063	1.12	17.4
	BQSquare	-0.069	0.90	14.8	-0.060	0.78	15.8
	RaceHorses	-0.101	1.53	14.4	-0.065	1.06	14.9
	Keiba	-0.111	1.85	14.1	-0.070	1.16	14.2
E	Vidyo1	-0.094	2.11	14.9	-0.049	1.10	15.4
	Vidyo3	-0.071	1.45	18.2	-0.053	1.08	19.2
	Vidyo4	-0.080	1.94	16.4	-0.050	1.22	16.9
	Johnny	-0.075	1.96	15.4	-0.047	1.22	15.6
	KristenAndSara	-0.080	1.71	18.9	-0.052	1.10	19.2
F	SlideEditing	-0.080	0.59	16.5	-0.072	0.53	17.2
	ChinaSpeed	0.001	-0.03	15.7	0.020	-0.25	16.1
	SlideShow	-0.094	1.12	15.9	-0.093	1.11	16.2
Average		-0.079	1.61	15.9	-0.052	1.04	15.9

Table 6. Hardware Implementation Performance Comparisons

Design	MaxSpeed [MHz]	HardwarCost [k gates]			Power [mW]	Cycle
		1D-DCT	TRF	Total		
Proposed	311	34.9	38.3	73.2	12.69	N+3/4N
[9]'s	418	15.9	59.6	75.5	14.21	2N
Primitive	311	199.7	60.4	260.1	42.64	2N

5. CONCLUSION

In this paper, the approximated DCT algorithms for HEVC PU RDO are proposed, which include the schemes of DCT matrix sparse butterfly decomposition, hardware friendly coefficient simplification, and high frequency domain prediction. According to the algorithm optimizations, we developed the associated VLSI DCT engine, in which a hardware reusing TRF was devised. Experiments revealed that our methods saved 15.9% Intra encoding time with 1.04% BDBR increase. With TSMC 90nm technology, our 73.2k-gate hardwired engine achieved the 311MHz clock speed under the worst working conditions (125°C, 0.9V), and the corresponding power dissipation is 12.7mW.

6. REFERENCES

[1] G.J. Sullivan and J.R. Ohm, "Recent developments in standardization of high efficiency video coding (HEVC)," in *SPIE Optical Engineering+ Applications*. International Society for Optics and Photonics, 2010, pp. 77980V–77980V.

[2] G.J. Sullivan, J.R. Ohm, W.J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Trans. Circuits and Systems for Video Tech*, 2012.

[3] D. Hein and N. Ahmed, "On a Real-Time Walsh-Hadamard/Cosine Transform Image Processor," *Electromagnetic Compatibility, IEEE Transactions on*, vol. 20, no. 3, pp. 453–457, 1978.

[4] Ashfaq Ahmed, Muhammad Usman Shahid, and Ata Ur Rehman, "N point DCT VLSI architecture for emerging HEVC standard," *VLSI Design*, vol. 2012, pp. 1–13, 2012.

[5] M. Masera, M. Martina, and G. Masera, "Adaptive Approximated DCT Architectures for HEVC," *IEEE Trans. Circuits and Systems for Video Tech*, vol. in Press, 2016.

[6] W.H. Chen, CH Smith, and S. Fralick, "A Fast Computational Algorithm For the Discrete Cosine Transform," *Communications, IEEE Transactions on*, vol. 25, no. 9, pp. 1004–1009, 1977.

[7] J. Zhu, Z. Liu, and D. Wang, "Fully pipelined DCT/IDCT/Hadamard unified transform architecture for HEVC Codec," in *2013 IEEE International Symposium of Circuits and Systems*, 2013, pp. 677–680.

[8] M. Jridi and P.K. Meher, "A Scalable Approximate DCT Architectures for Efficient HEVC Compliant Video Coding," *IEEE Trans. Circuits and Systems for Video Tech*, vol. in Press, 2016.

[9] J. Zhu, Z. Liu, and D. Wang, "Fast Prediction Mode Decision With Hadamard Transform Based Rate-Distortion Cost Estimation For HEVC Intra Coding," in *Image Processing (ICIP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1977–1981.

[10] ANIL K. JAIN, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.

[11] V. Sze, M. Budagavi, and G.J. Sullivan, *High Efficiency Video Coding (HEVC),Algorithms and Architectures*, Springer, 2014.

[12] L.L. Magoarou and R. Gribonval, "Flexible Multi-layer Sparse Approximations of Matrices and Applications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, 2015.

[13] Z. Liu, J. Zhou, D. Wang, and T. Ikenaga, "Register Length Analysis and VLSI Optimization of VBS Hadamard Transform in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 5, pp. 601–610, 2011.

[14] G. Bjontegard, "Calculation of average PSNR differences between RD-curves," *ITU-T VCEG-M33*, 2001.