J Real-Time Image Proc
DOI 10.1007/s11554-015-0549-8

SPECIAL ISSUE PAPER

Fast algorithms and VLSI architecture design for HEVC intra-mode decision

Xiaofeng Huang · Huizhu Jia · Binbin Cai · Chuang Zhu · Jie Liu · Mingyuan Yang · Don Xie · Wen Gao

Received: 14 May 2015 / Accepted: 25 November 2015

Abstract The emerging intra-coding tools of High Efficiency Video Coding (HEVC) standard can achieve up to 36 % bit-rate reduction compared to H.264/AVC, but with significant complexity increase. The design challenges, such as data dependency and computational complexity, make it difficult to implement a hardware encoder for real-time applications. In this paper, firstly, the data dependency in HEVC intra-mode decision is fully analyzed, which is cost by the reconstruction loop, the Most Probable Mode, the context adaption during Context-based Adaptive Binary Arithmetic Coding based rate estimation, and the Chroma derived mode. Then, several fast algorithms are proposed to remove the data dependency and to reduce the computational complexity, which include source signal based Rough Mode Decision, coarse to fine rough mode search, Prediction Mode Interlaced RDO mode decision, parallelized context adaption and Chroma-free Coding Unit (CU)/Prediction Unit (PU) decision. Finally, the parallelized VLSI architecture with CU reordering and Chroma reordering scheduling is proposed to improve the throughput. The experimental results demonstrate that the proposed intra-mode decision achieves 41.6 % complexity reduction with 4.3 % *Bjontegaard Delta Rate* (BDR) increase on average compared to the reference software, HM-13.0. The intra-mode decision scheme is implemented with 1571.7K gate count in 55 nm CMOS technology. The implementation results show that our design can achieve 1080p@60fps real time processing at 294 MHz operation frequency.

Keywords Data dependency · Computational complexity · Fast algorithm · HEVC intra-mode decision · VLSI architecture

X. Huang (✉) · H. Jia · B. Cai · C. Zhu · J. Liu · M. Yang · D. Xie · W. Gao
National Engineering Laboratory for Video Technology, Peking University, Beijing, China
e-mail: xfhuang@jdl.ac.cn

H. Jia
e-mail: hzjia@jdl.ac.cn

B. Cai
e-mail: bbcai@jdl.ac.cn

C. Zhu
e-mail: czhu@jdl.ac.cn

J. Liu
e-mail: jliu@jdl.ac.cn

M. Yang
e-mail: myyang@jdl.ac.cn

D. Xie
e-mail: donxie@jdl.ac.cn

W. Gao
e-mail: wgao@jdl.ac.cn

1 Introduction

The latest video coding standard HEVC, developed under the efforts of the Joint Collaborative Team on Video Coding (JCT-VC), has achieved substantial coding efficiency improvement over the H.264/AVC, but at the expense of increased computational complexity [1, 2]. The performance gain mainly comes from the new coding tools, such as larger Coding Tree Unit (CTU) (i.e., up to 64 × 64), recursive quad-tree structured CU split (i.e., from 64 × 64 down to 8 × 8), larger PU (i.e., from 64 × 64 down to 4 × 4), larger Transform Unit (TU) (i.e., from 32 × 32 down to 4 × 4), and more fine-grained intra-prediction modes [2]. In HEVC intra-coding, only square structure is specified for CU, PU, and TU. The PU partition

Published online: 16 December 2015

has the root at the CU level. The CU size is always equal to that of PU except the 8 × 8 CU, which supports both 8 × 8 and 4 × 4 PU [3]. The Residual Quad-Tree (RQT) partitions the CU residuals into multiple TUs, which also has the root at the CU level. The intra-frame coding capability in HEVC can achieve higher coding efficiency than that in previous standards, such as H.264/AVC and JPEG2000 [4]. Thus, this intra-only coding and decoding is very suitable for applications like digital still camera and satellite image transmission.

In HEVC intra-coding, each candidate partition of PU/TU/CU needs to pass through the complex intra-mode decision process to derive the Rate Distortion cost (RD-cost). Then, the partition with minimum RD cost is chosen as the best. The RDO-based intra-mode decision provides the best coding efficiency, but the computational complexity is increased dramatically. Besides the complexity, the data dependencies like the reconstruction loop make the HEVC intra-mode decision become challenging for efficient hardware implementation. These two bottlenecks should be resolved to implement a hardware encoder for real time applications.

To alleviate the complexity, several fast algorithms have been proposed, such as the fast RMD and the fast CU depth decision. The RMD is a preliminary mode decision process to select N most promising candidate modes from all 35 intra-prediction modes [5]. Jiang et al. [6] proposed a gradient-based fast RMD algorithm, where gradient directions and histogram were established based on Sobel edge operator. The principle is the same as in [7], which is proposed for fast intra-mode decision in H.264/AVC. Zhang et al. [8] proposed a progressive rough prediction mode search scheme to skip the unnecessary candidate prediction modes.

The fast CU depth decision in HEVC intra-mode decision can be classified into two categories. In the first category, works like [9, 10] straightforwardly eliminate the low-probability CU depth levels before the recursive CU split. In [9], the correlation between the current CTU and spatial neighbouring CTUs is fully utilized to determine the CU depth search range. Work [10] uses the variance to determine the CU depth, where a threshold is empirically derived. In the second category, works like [8, 11] skip certain CUs during the recursive CU split when predefined conditions are satisfied. Work [8] skips the large CU based on the block structures of its sub-CUs, and work [11] skips the further split based on the RD cost correlation between the parent CU and its partial sub-CUs. These fast algorithms achieve the computation complexity reduction with insignificant coding efficiency degradation. However, most of the works are not hardware oriented and they cannot be implemented easily in hardware.

Besides the complexity, the strong data dependencies like the reconstruction loop challenge the hardware implementation for HEVC intra-mode decision. Until now, several hardware architectures of HEVC intra-mode decision have been proposed. Module level architectures, such as Intra-Prediction (IP) [12, 13] and DCT [14, 15] have been proposed to achieve the super-HD real-time processing. Six-stage pipeline intra-mode decision VLSI architecture is exclusively designed for 4 × 4 block in Li and Shi [16]. A single-chip HEVC 8192 × 4320p encoder chip is implemented in Tsai et al. [17]. The complex CU 8 × 8 and PU 4 × 4 are eliminated for high-resolution applications. The fully parallelized (64 × 64 intra-CU, 32 × 32 intra-CU and 16 × 16 intra-CU are parallelized) VLSI architecture is adopted to meet the 8192 × 4320p@30fps real-time processing. An HEVC intra-encoder with source texture based CU/PU mode pre-decision is proposed in Zhu et al. [18]. The CU/PU candidate reduction is decided by the RD cost estimation from the source image texture. Two CU/PU mode candidates, one CU candidate from 32 × 32 and 16 × 16 CU, and one PU candidate from 8 × 8 and 4 × 4 PU, will be decided by the simplified RD cost estimation algorithm. In the scheduling, the rough search, fine search and reconstruction are sequentially processed to decide the best prediction mode. Parallelism and pipelining strategies are adopted in these architecture designs, but the data dependency in HEVC intra-mode decision is not fully taken into consideration.

Unlike the emerging HEVC intra-encoder, the data dependency in H.264/AVC has been extensively studied and resolved. The contributions for solving the data dependency problem in H.264/AVC can be briefly classified into two categories. In the first category, the neighbouring original samples, instead of reconstructed ones, are used as the reference data for intra-prediction [19, 20]. However, this data dependency optimization sacrifices too much coding efficiency. In the second category, the efficient block scheduling is utilized to improve the pipeline efficiency [21, 22]. In [21], an interlaced block reordering scheme is proposed to allow the fully parallel operation of the mode decision and reconstruction. In [22], the 4 × 4 block reordering strategy and the interlaced scheduling are proposed. However, these techniques cannot be directly applied to the HEVC intra-mode decision due to the stronger data dependency of the latter.

In this paper, we focus on hardware-oriented fast algorithms and the corresponding VLSI architecture design for HEVC intra-mode decision. Firstly, the data dependency in HEVC intra-mode decision is fully analyzed, which is cost by the reconstruction loop, the MPM, the context adaption during CABAC based Rate estimation (CABACR), and the Chroma derived mode. The CABACR, which has been adopted in HEVC reference software, is to calculate the

rate R for RDO mode decision [23]. Then, hardware-oriented fast algorithms and VLSI architecture are proposed for HEVC intra-mode decision. In fast algorithm part, source signal based RMD and mode-dependent R_{mode} scheme allow the CTU-level pipeline scheduling for RMD and RDO mode decision. Besides, coarse to fine rough mode search scheme lowers the computational complexity in RMD. PMI RDO mode decision resolves the reconstruction loop dependency for CU with split TU, and parallelized context adaption resolves the data dependency during CABACR, respectively. The proposed Chroma-free CU/PU decision resolves the data dependency incurred by the Chroma derived mode and reduces the luminance prediction modes for RDO mode decision. In hardware part, the parallelized VLSI architecture with CU reordering and Chroma reordering scheduling is proposed to improve the throughput. As a result, the proposed intra-mode decision achieves 41.6 % complexity reduction with 4.3 % BDR increase on average. The intra-mode decision scheme is implemented with 1571.7 K gate count in 55 nm CMOS technology. The implementation results show that our design can achieve the 1080p@60fps real time processing at 294 MHz operation frequency, which outperforms the state-of-the-art work [18].

The remainder of this paper is organized as follows. Section 2 presents an overview of intra-mode decision, in which the data dependency is fully analyzed. Section 3 describes the proposed fast intra-mode decision algorithm, and Sect. 4 proposes our VLSI architecture. Comprehensive experiments have been carried out, and the results are illustrated in Sect. 5. Finally, Sect. 6 concludes the work in this paper.

2 Data dependency analysis

2.1 Intra-mode decision overview

The intra-mode decision flow in HEVC reference software is shown in Fig. 1. The intra-mode decision can be split into two phases, which are RMD and Rate Distortion Optimization (RDO) mode decision. The RMD is a preliminary mode decision process to select N most promising candidate modes from all 35 intra-prediction modes, and the RDO mode decision decides the best intra-prediction mode and PU/TU/CU structure.

In the RMD phase, the N most promising candidate modes are chosen from all 35 intra-prediction modes using the following cost function (1):

$$C_{\text{RMD}} = D_{\text{Had}} + \lambda_{\text{RMD}} R_{\text{mode}} \tag{1}$$

In (1), D_{Had} is the absolute sum of Hadamard transformed residuals for a PU, R_{mode} represents the entropy coding bits of the prediction mode, and λ_{RMD} is the Lagrange multiplier for the tradeoff between D_{Had} and R_{mode} in RMD. The residuals used for D_{Had} calculation are the difference between source signal and prediction samples. The prediction samples are generated using the neighbouring reconstructed samples as the reference data. The R_{mode} is calculated based on the MPM (the best modes of spatial neighbouring PUs) [24] and the context state of the Syntax Element (SE) *prev_intra_luma_pred_flag* [3, 23].

In the second phase, the RDO mode decision is to decide the best intra-prediction mode and PU/TU/CU structure. The cost function in this phase is shown as (2).

$$C_{\text{RD}} = D + \lambda R \tag{2}$$

In (2), the distortion D is the sum of squared errors between source signal and reconstructed samples. The bitrate R is the estimated coding bits of the quantized coefficient and header syntax by the adaptive CABAC rate estimation method [23]. The Lagrange parameter λ determines the tradeoff between distortion D and the rate R.

A detailed flow in RDO mode decision is described in Fig. 2, which consists of luminance prediction mode decision, TU depth decision (RQT), Chroma prediction mode decision, and PU/CU structure decision. The entering

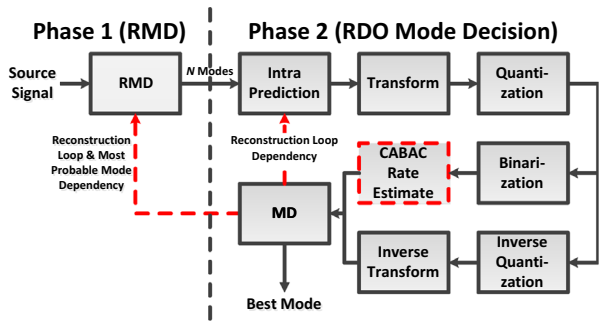


Fig. 1 HEVC intra-mode decision flow

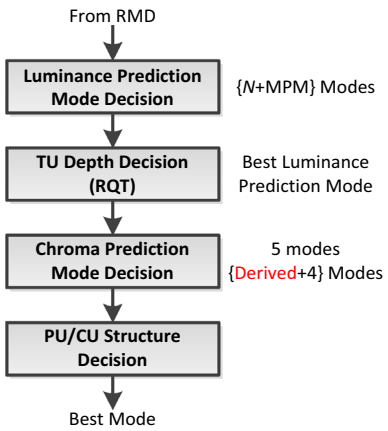


Fig. 2 Detailed flow in RDO mode decision

prediction modes are listed at the right column. In luminance prediction mode decision, the N candidate modes decided from RMD correlate with PU size. {8, 8, 3, 3, 3} is assigned to 4 × 4, 8 × 8, 16 × 16, 32 × 32, and 64 × 64 PU, respectively. Additionally, the MPM will be added into the candidates when they are not included in the N candidate modes [24]. The RQT is used to decide the best TU structure using the best luminance prediction mode [3]. Five prediction modes including the derived mode (i.e., the corresponding luminance PU prediction mode) are decided to get the best Chroma prediction mode. For PU/CU structure decision, the best structure is decided. All the decision processes in Fig. 2 are all made according to the RD-Cost (C_{RD}) as described in (2).

2.2 Data dependency analysis

The data dependency in HEVC intra-mode decision mainly comes from four aspects, which contains the reconstruction loop, MPM, the context adaption, and the Chroma derived mode. These dependencies degrade the pipeline efficiency and impede the real-time hardware implementation [21].

As shown in Fig. 1, there is a feedback from RDO mode decision to RMD. It is introduced by the reconstruction loop and the MPM dependency. In RMD phase, the cost function (1) is calculated based on neighbouring reconstructed samples and MPM. However, these can be only acquired in RDO mode decision phase. A pipeline scheduling with the dependency is illustrated as in Fig. 3. The hypothesis in Fig. 3 is that the RMD and RDO mode decision are designed in pipelining. In Fig. 3a, two neighbouring PUs are used for illustration. The pipeline scheduling (space–time diagram [25]) is shown in Fig. 3b. The horizontal coordinate is time, and the vertical coordinate is pipeline stage. In the pipeline design, the first stage is the RMD, and the second stage is the RDO mode decision (MD). Due to the data dependency, the RMD of the PU_1 block will not start until the PU_0 block finishes the RDO MD. Thus, the RMD is idle during the RDO MD period, and the RDO MD is idle during the RMD period.

As shown in Fig. 1, the reconstruction loop dependency also exists in the RDO mode decision, especially when the CU is split into sub-TUs (such as RQT and CU 64 × 64). The reconstruction loop dependency comes from the fact that the intra-prediction is applied for each TU sequentially. The pipeline scheduling for CU with split TU is illustrated in Fig. 4. It is assumed that the RDO mode decision is designed with three-stage pipeline structure. The pipeline scheduling for CU with split TU is shown in Fig. 4b. Due to the reconstruction loop dependency, the RDO mode decision for each intra-prediction mode (A, B, C, etc.) must be executed on TU sequentially.

As shown in Fig. 1, the context adaption in CABAC rate estimate module also incurs a serious data dependency. The data dependency comes from the regular bins (bins coded with adaptive context) [23]. The context state should be updated before the next bin can use. The context adaption dependency will prolong the processing time in RDO mode decision, which causes the throughput degradation.

The Chroma derived mode is the best prediction mode of the corresponding luminance PU. The Chroma derived mode can be obtained only after its corresponding luminance PU prediction mode is decided.

These data dependencies challenge the pipeline efficiency and impede the efficient hardware implementation. Besides the dependencies, the abundant modes to be decided in Fig. 2 aggravate the computational complexity. These problems must be resolved to achieve the 1080p@60fps real time processing for HEVC intra-mode decision. The 1080p@60fps real time processing represents that the design can process sixty 1920 × 1080 Frames Per Second (fps). In the next section, several hardware friendly fast algorithms are proposed to remove data dependency and reduce complexity. Source signal-based RMD and mode-dependent R_{mode} scheme remove the feedback from RDO mode decision to RMD in Fig. 3, which allow the CTU-level pipeline scheduling for RMD and RDO mode decision. Besides, coarse to fine rough mode search scheme decreases the computational complexity in RMD. PMI RDO mode decision resolves the reconstruction loop dependency for CU with split TU in Fig. 4, and parallelized context adaption resolves the data dependency during CABACR, respectively. The proposed Chroma-free CU/PU decision resolves the data dependency incurred by

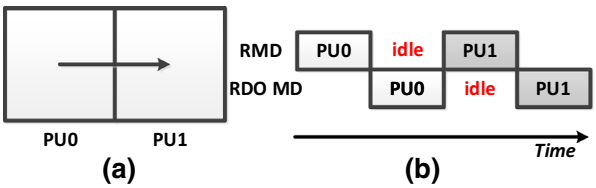


Fig. 3 Pipeline scheduling with data dependency in intra-mode decision. **a** Two neighboring PUs. **b** Pipeline scheduling for PU_0 and PU_1

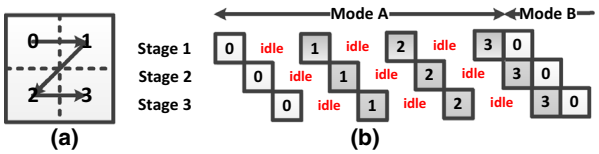


Fig. 4 Pipeline scheduling with data dependency in RDO mode decision. **a** CU with split TU. **b** Pipeline scheduling for CU with split TU

the Chroma derived mode and reduces the prediction modes in luminance prediction modes decision in Fig. 2.

3 Proposed fast algorithms

In this section, several hardware friendly fast algorithms are proposed to remove data dependency and to reduce computational complexity, which include coding tools analysis, fast RMD algorithm (contains source signal based RMD, mode dependent R_{mode} scheme and coarse to fine rough mode search), PMI RDO mode decision, parallelized context adaption and Chroma-free CU/PU decision.

3.1 Coding tools analysis

In this section, the coding tools of RQT, TU 32 × 32, and CU 64 × 64 are analyzed. Through analysis, it can be concluded that these coding tools can be all disabled to decrease the computational complexity, to increase the pipeline efficiency, and to save the chip area [18, 26]. This achieves 9.3 % complexity reduction with 1.4 % BDR increase on average as shown in Table 1.

The performance comparison in this paper is evaluated in terms of the change of average BDR [27]. The performance gain or loss is measured with respect to HM-13.0. The test configuration is "All Intra-Main". RDO Quantization (RDOQ) and transform skip coding tools are disabled. QP values of 22, 27, 32 and 37, and sequences recommended by JCT-VC are used for evaluation.

3.1.1 RQT

RQT partitions CU residuals into TUs, which represents that a 2 N × 2 N CU can be recursively split into sub-TUs [2]. As shown in Fig. 2, the RQT is processed using the best luminance prediction mode. The sequential processing of the luminance prediction mode decision and RQT not only burdens the computational complexity, but also prolongs the processing time in RDO mode decision. The pipeline scheduling for RQT (CU with split TU) is in detail analyzed in Fig. 4. It shows that the RQT will degrade the pipeline efficiency.

In order to decrease the computational complexity and increase the pipeline efficiency, the RQT is disabled in our design. This means that the TU size is always equal to the corresponding PU size [18]. As in Table 1, the BDR is increased by 0.4 % on average.

3.1.2 TU 32 × 32

Flexible TU size (from 4 × 4 to 32 × 32) is introduced in HEVC to improve the coding efficiency. As shown in Table 1, the BDR is increased by 0.9 % on average by disabling the coding tool of TU 32 × 32. The coding performance shows significant degradation for some sequences having flat regions, such as *Kimono* (6.1 %) and *Johnny* (3.4 %). However, we still disable the coding tool of TU 32 × 32 for the consideration of saving the chip area. In [26], a detailed analysis of the transpose memory in 2-D (I)DCT is presented. The chip area of the 32 × 32 transpose memory is about three times larger than that of 16 × 16. The disabling coding tool of TU 32 × 32 is a tradeoff between the coding efficiency and the chip area. In our architecture, TU 32 × 32 is disabled, and the maximum TU size is set to 16 × 16.

3.1.3 CU 64 × 64

For CU 64 × 64, the intra-prediction must be applied on each TU sequentially as shown in Fig. 4b. This will degrade the pipeline efficiency. In our architecture, CU 64 × 64 is disabled with 0.1 % BDR increase on average as shown in Table 1. The disabling coding tool of CU 64 × 64 will reduce the computational complexity [18]. Besides, the chip area consumed for CU 64 × 64 can be saved.

Table 1 Coding tools performance (vs. HM-13.0)

	RQT OFF (%)	TU 32 × 32 OFF (%)	CU 64 × 64 OFF (%)	RQT + TU 32 × 32 + CU 64 × 64 OFF (%)	Complexity reduction (%)
Class A	0.4	0.4	0.0	0.8	8.1
Class B	0.3	2.0	0.1	2.4	9.7
Class C	0.4	0.4	0.0	0.8	9.5
Class D	0.3	0.2	0.0	0.5	9.5
Class E	0.6	1.8	0.2	2.6	9.0
Class F	0.5	0.4	0.1	1.1	9.8
Average	0.4	0.9	0.1	1.4	9.3

3.2 Fast RMD algorithm

The RMD selects N most promising candidate modes from all 35 intra-prediction modes. However, the feedback from RDO mode decision to RMD degrades the pipeline efficiency as shown in Fig. 3b. To resolve the problem, source signal based RMD and mode dependent R_{mode} scheme are proposed. Besides, coarse to fine rough mode search algorithm is proposed to decrease the computational complexity in RMD.

The feedback is caused by the reconstruction loop and MPM. Instead of using the neighboring reconstructed samples, using the neighboring source signal for RMD will break down the reconstruction loop dependency.

The calculation of R_{mode} in RMD is in detail described in Sect. 2. To eliminate the MPM dependency, mode dependent R_{mode} scheme is proposed. Table 2 shows the percentages of best intra-prediction mode for *Traffic* in class A, *BasketballDrive* in class B, and *BlowingBubbles* in class D at different QPs. In Table 2, we can see that planner (0) and DC (1) modes have larger probability to be selected as the best prediction mode, and then followed by horizontal (10) and vertical prediction (26) modes. The mode in the bracket is the maximum-percentage mode in the remaining ones. The maximum-percentage mode is highly correlated with the input content, and the percentage is much smaller compared to that of planar/DC mode. According to this statistical analysis, a mode dependent R_{mode} method is proposed as shown in Table 3. Fewer bits are assigned to the high percentage modes, and more bits are assigned to the rarely chosen ones. For prediction mode 0 and 1, R_{mode} is assigned to 0. For prediction mode 10 and 26, R_{mode} is assigned to 1. For the remaining modes, R_{mode} is assigned to 5. The 0, 1 and 5 are derived according to the bits consumed by the SEs of *mpm_idx* (1-bit) and *rem_intra_luma_pred_mode* (5-bit), respectively [23, 28].

By using the source signal based RMD and the mode dependent R_{mode} scheme, the feedback from RDO mode decision to RMD can be completely removed. The updated intra-mode decision flow can be illustrated in Fig. 5. The feedback from RDO mode decision to RMD can be eliminated as shown in Fig. 5a. In Fig. 5b, the CTU-level seamless pipeline scheduling can be applied for RMD and RDO mode decision, which increases the pipeline efficiency by about 50 % compared to Fig. 3b. As shown in Table 4, the BDR is increased by 0.2 % on average with the proposed schemes.

Table 2 Percentages of best intra-prediction mode

Sequence	QP	Mode (%)				
		Planar	DC	Hor.	Ver.	Max. Others
Traffic	22	18.7	20.0	8.0	4.7	6.6 (11)
	37	25.6	14.3	7.2	6.7	6.4 (11)
BasketballDrive	22	19.7	15.8	20.2	7.4	4.5 (11)
	37	24.8	10.7	15.1	9.1	5.0 (11)
BlowingBubbles	22	22.8	19.3	2.6	3.1	6.7 (24)
	37	27.9	16.2	4.3	6.2	2.5 (22)
Average		23.3	16.1	9.5	6.2	5.3

Table 3 Mode dependent R_{mode}

Mode	R_{mode}
0, 1	0
10, 26	1
2–9, 11–25, 27–34	5

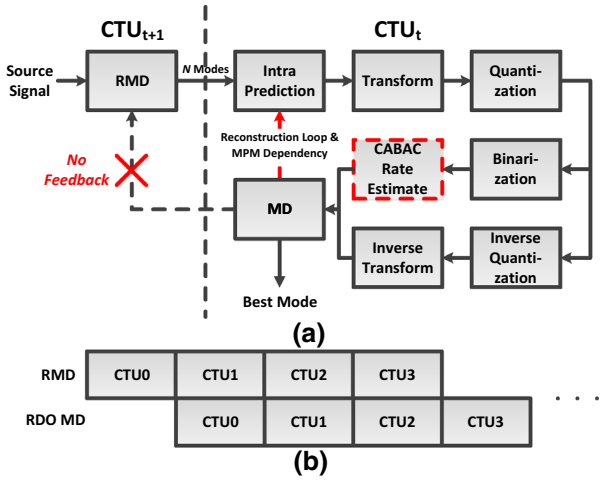


Fig. 5 The updated intra-mode decision flow. **a** No feedback from RDO mode decision to RMD. **b** The CTU level pipeline scheduling for RMD and RDO mode decision

Instead of using the cost function (1), the cost function (3) is used for RMD. Sum of Absolute Difference (SAD) is used as the criterion for the distortion calculation, and R_{mode} is calculated based on Table 3. The chip area of the Hadamard transform, about 16.9 K gate count for an 8×8 Hadamard transform, can be saved. As shown in Table 4, the BDR is increased by 0.4 % on average with the substitution from SATD to SAD.

$$C_{\text{RMD}} = \text{SAD} + \lambda_{\text{RMD}} R_{\text{mode}} \tag{3}$$

In addition, to decrease the computational complexity in RMD, coarse to fine rough mode search algorithm is proposed. In the coarse search step, angular prediction modes of {2, 6, 10, 14, 18, 22, 26, 30, and 34} are processed based on (3). The best prediction mode (BP_1) will be decided from these modes. For example, supposing mode 10 to be the BP_1. In the fine search step, the four nearest neighboring prediction modes of BP_1 (for the example they are

Table 4 Fast RMD coding performance comparison (vs. HM-13.0)

	Proposed (%)	DR (%)	SAD (%)	CF (%)	[8] (%)	[6] (%)
Class A	1.2	0.2	0.5	0.5	0.3	0.80
Class B	1.1	0.2	0.4	0.5	0.3	0.80
Class C	1.3	0.2	0.3	0.8	0.6	0.70
Class D	1.2	0.1	0.4	0.7	0.4	0.80
Class E	1.6	0.4	0.4	0.7	0.7	1.60
Class F	1.0	0.3	0.3	0.5	0.3	1.20
Average	1.2	0.2	0.4	0.6	0.4	1.00

{8, 9, 11, 12}) and the planar (0), DC (1) modes will be decided in RMD.

Totally 15 prediction modes will be decided in RMD, with 9 modes in the coarse search step and 6 modes in the fine search step. The N most promising candidate prediction modes will be selected from the 15 prediction modes. The proposed algorithm has 0.6 % BDR increase as in Table 4. The coding performance of [8] is better than our proposed algorithm. It is because more prediction modes (about 21 prediction modes) are decided in RMD in [8]. Compared to [6], our algorithm achieves better coding performance. It is because that the gradient based fast RMD algorithm is too coarse for the RMD.

Integrating all these proposed fast RMD algorithms into HM-13.0, the BDR will be increased by 1.2 % (the second column, Proposed) on average as shown in Table 4. The DR means the data Dependency Removal algorithms, which include source signal based RMD and mode dependent R_{mode} scheme. The SAD represents substitution from SATD to SAD. The CF is the proposed Coarse to Fine rough mode search algorithm. The BDR is increased by 0.2, 0.4, and 0.6 % for the DR, SAD and CF, respectively.

3.3 PMI RDO mode decision

For CU with split TU, the reconstruction loop dependency degrades the pipeline efficiency as shown in Fig. 4b. Due to the maximum supported TU size is 16 × 16 in our architecture, the CU 32 × 32 will be split into four 16 × 16 TUs. To resolve the data dependency problem for CU with split TU (i.e., CU 32 × 32); a PMI RDO mode decision is proposed.

As shown in Fig. 6, the PMI RDO mode decision is proposed. In Fig. 6a, the pipeline scheduling for CU with split TU is presented, which is the same as in Fig. 4a. For each prediction mode, the RDO mode decision should be applied on each TU sequentially. To improve the throughput, a PMI pipeline scheduling is proposed as shown in Fig. 6b. For each TU, all the prediction modes will be executed seamlessly. Compared to Fig. 6a, it achieves the pipeline efficiency improvement.

The overhead of the proposed PMI RDO mode decision scheme is that all the quantized coefficients for each prediction mode should be stored, which will require much more chip area. To save the chip area, the RDO mode decision under the best prediction mode is undergoing after the best mode is decided by the PMI RDO mode decision. In this way, only the quantized coefficients for the best prediction mode need to be stored.

3.4 Parallelized context adaption

In CABAC rate estimation process, the context state should be updated before the next bin can use it. The parallelism architecture can be applied for bins coded by different contexts. However, it is difficult to parallelize the bins

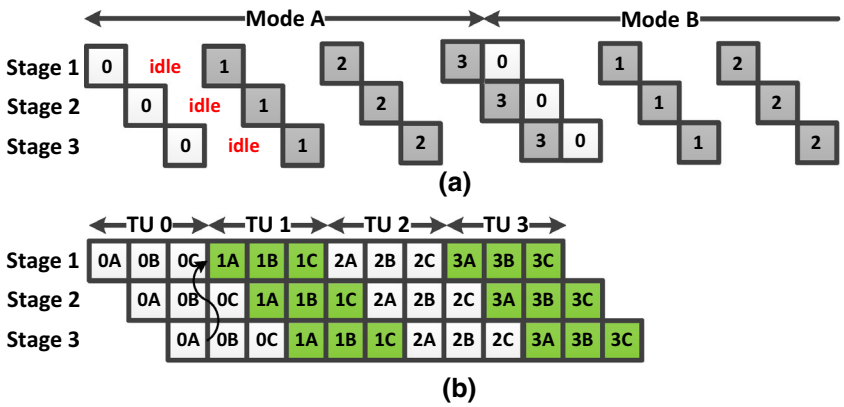


Fig. 6 Proposed PMI RDO mode decision. **a** Pipeline scheduling for CU with split TU. **b** The PMI pipeline scheduling for CU with split TU

coded by the same context. Compared to other SEs, the *coeff_abs_level_greater1_flag* and the *sig_coeff_flag* are the bottleneck during the CABAC rate estimation [29, 28]. In this section, parallelized context adaption algorithm is proposed to resolve the context adaption problem.

The context selection for the *coeff_abs_level_greater1_flag* depends on the number of trailing ones and the number of coefficient level larger than 1 as shown in Table 5 [29]. A fixed context selection is proposed to resolve the data dependency as shown in Table 5. The context 1 is assigned to the first two bins, and context 2, 3, 0 are assigned to the remaining bins, respectively. Maximum 2 bins will utilize the same context by this method.

For *sig_coeff_flag*, maximum of 16 bins in a Coefficient Group (CG) will utilize the same context when both the right and lower CGs are significant [29]. Eight cycles are needed when the throughput is two bins/cycle. To improve the throughput, a fast context adaption algorithm is proposed. The *sig_coeff_flag* bins are organized in pairs to be coded. The input state of each pair is the state at the input of the CG. The context state output to the next CG is the updated state of the last coded bin pair. At maximum two bins are dependent by the scheme.

By the proposed parallelized context adaption algorithm, the CABAC rate estimation in a CG can be fully parallelized. The proposed fast context adaption causes 0.5 % BDR increase on average as shown in Table 6. Our algorithm achieves the best coding performance compared to work [17, 30]. Compared to [17], our work updates the context in real time, which leads to a better coding performance. In [30], a coefficient-level rate estimation scheme is proposed with the fixed context state. It is noted that the actual CABAC is not included here and should be handled separately.

3.5 Chroma-free CU and PU decision

As illustrated in Sect. 2, the Chroma derived mode can be obtained only after its corresponding best luminance PU prediction mode is decided. In Fig. 2, the best CU/PU structure is decided based on the C_{RD} of the luminance and Chroma prediction mode decision. To improve the pipeline efficiency, the C_{RD} of the Chroma component is removed from the CU/PU structure decision. Besides, only the derived mode is processed for the Chroma component. And the luminance prediction modes for RDO mode decision are fixed. With all these schemes, 1.2 % BDR is increased.

Table 5 Context selection for *coeff_abs_level_greater1_flag*

	Context	Description
HEVC Standard [29]	1	Initial—no trailing ones
	2	1 Trailing one
	3	2 or More trailing ones
	0	1 or More larger than 1
Fast Adaption	1	From 1 to 2 bins
	2	From 3 to 4 bins
	3	From 5 to 6 bins
	0	From 7 to 8 bins

Table 6 Fast rate estimation performance comparison (vs. HM-13.0)

	Proposed	[17]	[30]
Class A	0.5	1.0	2.6
Class B	0.6	1.3	2.9
Class C	0.5	1.2	2.0
Class D	0.5	1.2	1.8
Class E	0.6	1.2	2.7
Class F	0.4	0.9	1.2
Average	0.5	1.1	2.2

In Fig. 2, the PU/CU structure decision should take both the luminance and Chroma component into consideration. The expression can be illustrated as (4). In (4), D_L and R_L represent the distortion and rate of luminance component; D_C and R_C represent the distortion and rate of Chroma component, λ is the Lagrange parameter, respectively.

$$C_{CU} = (D_L + D_C) + \lambda \cdot (R_L + R_C) \tag{4}$$

To remove the data dependency, the C_{CU} of the Chroma component is removed from the CU/PU structure decision. The updated cost function is shown as (5).

$$C_{CU} = D_L + \lambda \cdot R_L \tag{5}$$

The BDR is increased by 0.2 % on average when only the luminance component is considered for PU/CU structure decision.

As shown in Fig. 2, five prediction modes including the derived mode should go through the RDO mode decision to get the best Chroma prediction mode. In Table 7, the percentages of the best Chroma prediction mode are tabulated for *Traffic* in class A, *BasketballDrive* in class B, and *BlowingBubbles* in class D at different QPs. Compared to other modes, the percentage of the derived mode is the highest. This is because less header bits are coded for derived mode. To decrease the computational complexity, we directly assign the derived mode as the best Chroma prediction mode. The Chroma component needs to undergo the RDO mode decision process to get the standard complying reconstructed samples and quantized coefficients. The BDR is increased by 0.6 % with this method.

As shown in Fig. 2, the MPM will be added into the candidates when they are not included in the N candidate modes. The uncertain number of prediction modes causes the difficulty in the RDO mode decision pipeline

Table 7 Percentages of best chroma prediction mode

Sequence	QP	Mode (%)					
		Derived	Planar	DC	Hor.	Ver.	34
Traffic	22	62.3	11.5	10.3	7.8	5.6	2.4
	37	87.6	3.2	2.4	3.1	2.1	1.6
BasketballDrive	22	76.1	6.3	3.4	6.4	5.8	2.0
	37	81.2	4.0	1.9	5.3	3.8	3.8
BlowingBubbles	22	42.8	29.4	10.0	7.0	9.1	1.7
	37	69.2	19.5	4.0	2.7	3.3	1.2
Average		69.9	12.3	5.3	5.4	5.0	2.1

scheduling. To resolve the problem, the number of luminance prediction mode is fixed as shown in Table 8. The mode number N is derived according to the RDO mode decision pipeline scheduling which will be described in Sect. 4. It is noted that when the mode from the RMD is the same as the MPM, the following sub-optimal mode will be added into the candidates until reaching the fixed mode number N. The BDR is increased about 0.4 % on average with the fixed luminance prediction mode number.

By using the proposed fast algorithm in this section, the RDO mode decision flow in Fig. 2 can be updated as shown in Fig. 7. The RDO mode decision flow is divided into two stages. In the first stage, best luminance prediction mode and PU/CU structure are decided. In this way, the best Chroma prediction mode can also be derived. In the second stage, the Chroma RDO pass is executed to get the standard complying reconstructed samples and quantized coefficients. These two stages are pipelined at 32 × 32 CU level, and the detailed pipeline scheduling will be described in Sect. 4.

Integrating all the fast algorithms in this section, the BDR will be increased by 1.2 % on average.

In summary, several fast algorithms are proposed to remove data dependency and reduce computational complexity, including the coding tools disabling (1.4 %), the fast RMD (1.2 %), the PMI RDO mode decision, the parallelized context adaption (0.5 %), and the Chroma-free PU/CU decision (1.2 %). Integrating all these fast algorithms into HM-13.0, the proposed intra-mode decision scheme achieves 41.6 % complexity reduction with 4.3 % BDR increase. In the next section, the parallelized RDO mode decision VLSI architecture and efficient pipeline scheduling are proposed to achieve the 1080p@60fps real time processing.

Table 8 Fixed luminance prediction mode number

Size	Mode number (N)	Description
4 × 4	8	2MPM + 6 Modes from RMD
8 × 8	4	2MPM + 2 Modes from RMD
16 × 16	4	2MPM + 2 Modes from RMD
32 × 32	2	1MPM + 1 Modes from RMD

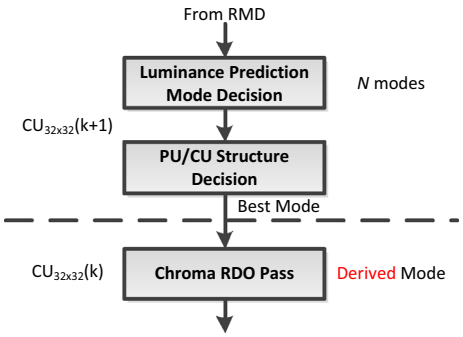


Fig. 7 Updated flow in RDO mode decision

4 RDO mode decision VLSI architecture

The CTU-level pipeline architecture is applied for RMD and RDO mode decision as shown in Fig. 5. The data dependency in RMD has been completely removed. The parallel and pipeline architecture can be applied in RMD VLSI architecture design [13]. In this section, we concentrate on VLSI architecture design for the RDO mode decision. A parallel and pipelined VLSI architecture is proposed in this section to achieve the 1080p@60fps real-time processing.

4.1 RDO mode decision VLSI architecture

Parallel architecture is imperative for the RDO mode decision design to achieve the 1080p@60fps real time processing. In our architecture, the processing latency of RDO mode decision for a 4 × 4 PU is 32 cycles. 251 MHz operation frequency is required to process all 4 × 4 PU in a CTU. The serial processing of all CU/PU modes will require 680 MHz operation frequency. This is far beyond the processing ability for 55 nm CMOS technology. To resolve the problem, 4 × 4 PU and the remaining PUs (including 8 × 8/16 × 16/32 × 32 PU) are in parallel processed.

The VLSI architecture of the RDO mode decision is designed as shown in Fig. 8. The core modules include the 4 × 4 TU RDO mode decision (RDO-4), the 8 × 8/16 × 16 TU RDO mode decision (RDO-8/16), and the RDO top (RDO-top). The 4 × 4 PU is processed in RDO-4 module; the 8 × 8/16 × 16/32 × 32 PU is processed in RDO-8/16 module. In the architecture, the RDO-4 and RDO-8/16 modules are both implemented with the parallelism of 16 pixels. In RDO-8/16 module, two pixel rows are processed for 8 × 8 TU, and one pixel row is processed

for 16 × 16 TU, respectively. In RDO-4 module, a 4 × 4 block is processed for 4 × 4 TU.

Pipeline architectures are adopted in these two RDO mode decision modules. In RDO-8/16 module, 6-stage pipeline architecture is adopted. Intra-prediction (IP) module is in the first stage to generate the prediction samples. The Horizontal DCT (DCTH) and Vertical DCT (DCTV) are in the second and third stage, respectively. Quantization (Q) is merged into DCTV module, and generates the quantized coefficients. Then, the pipeline architecture is divided into 2 branches. The binarization (BINGEN) and the IQ/IDCTV module are in the fourth stage. The IDCTH and CABACR modules are at the same stage, which become the fifth stage. The CABACR module calculates the bit-rate (R). The IDCTV module generates the reconstructed samples. The TUMD module is in sixth stage, which calculates the C_{RD} and decides the best prediction mode for each PU. Implementing the architecture, each stage will need 10 cycles to process an 8 × 8 block, and will need 22 cycles to process a 16 × 16 block, respectively.

A 25-stage fully pipelined architecture is implemented for RDO-4 module. Due to the parallelism of 16 pixels, a 4 × 4 block can be processed in each pipeline stage. The proposed architecture for 2D DCT (including DCTH and DCTV) in RDO-4 module is shown as in Fig. 9. In Fig. 9a, the four-point integer 1-D DCT is illustrated, which is the basic processing unit for the 2-D DCT. SA is the shift adder, which can be configured to support $(b(i) \ll 6) + (b(i) \ll 4) + (b(i) \ll 1) + b(i)$ $(83 \times b(i))$ and $(b(i) \ll 5) + b(i) \ll 2$ $(36 \times b(i))$. Traditional 2-D DCT architecture is shown as in Fig. 9b, where a transpose buffer is needed to transpose the data between the DCTH and DCTV [14]. The proposed 2-D DCT architecture is shown as in Fig. 9c. The 16-pixel parallelism makes the transpose buffer in Fig. 9b become the interconnected bus as shown in Fig. 9c.

The RDO-top includes the CUMD module and the buffers. The CUMD is to decide the best PU/CU structure. The buffers are all colored as gray blocks in Fig. 8. The CS buffer 0/1 (store the context state for CABACR), coding information (info.) buffer 0/1 (storing the coding information including the best CU/PU structure and prediction mode), quantized coefficient (Q-ed Coeff.) buffer 0 (storing the quantized coefficient of the best prediction mode) and reference (Ref.) pixels buffer 0/1 (storing the reference pixels for intra-prediction) are all implemented with registers to improve the throughput. The remaining buffers are implemented with SRAMs. These buffers are described as below.

- Source Samples Buffer, Modes Buffer and Upper CTU Line Buffer. Source samples buffer (128-b width, 384 length) stores the source samples, and modes buffer (32-b width, 596 length) stores the rough decided modes from RMD. The upper CTU line buffer (32-b width, 960 length) stores the reconstructed samples of the upper CTU row.
- Prediction Buffer 0/1, and source Buffer 0/1. The buffers are used to transfer the source and prediction samples from IP to TUMD for C_{RD} calculation. The SRAM of prediction (source) buffer 0 are both (128b-

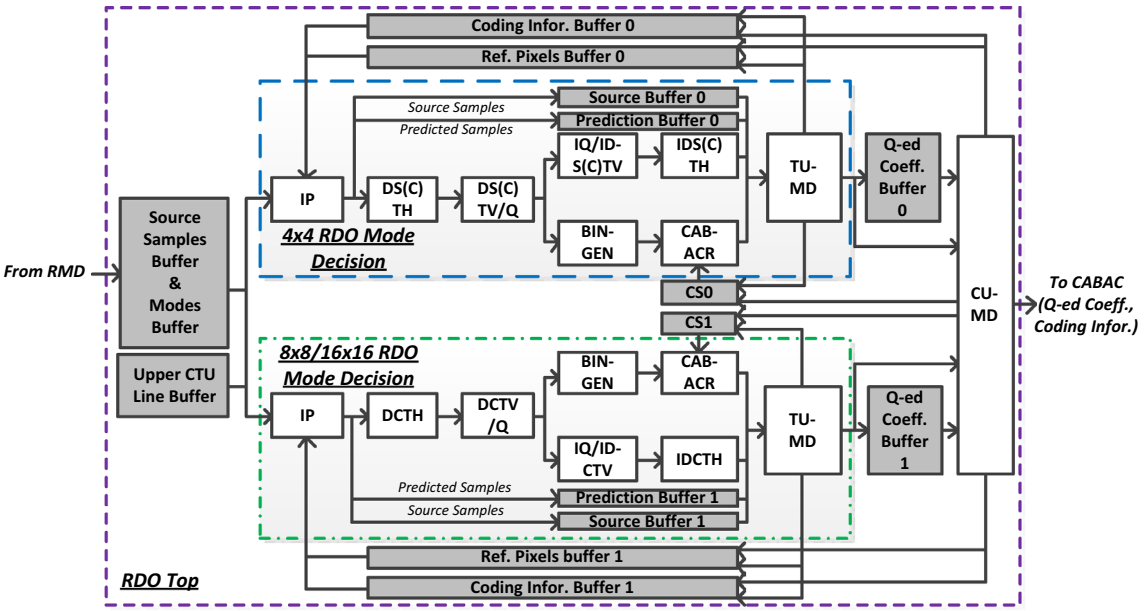


Fig. 8 Proposed RDO mode decision VLSI architecture

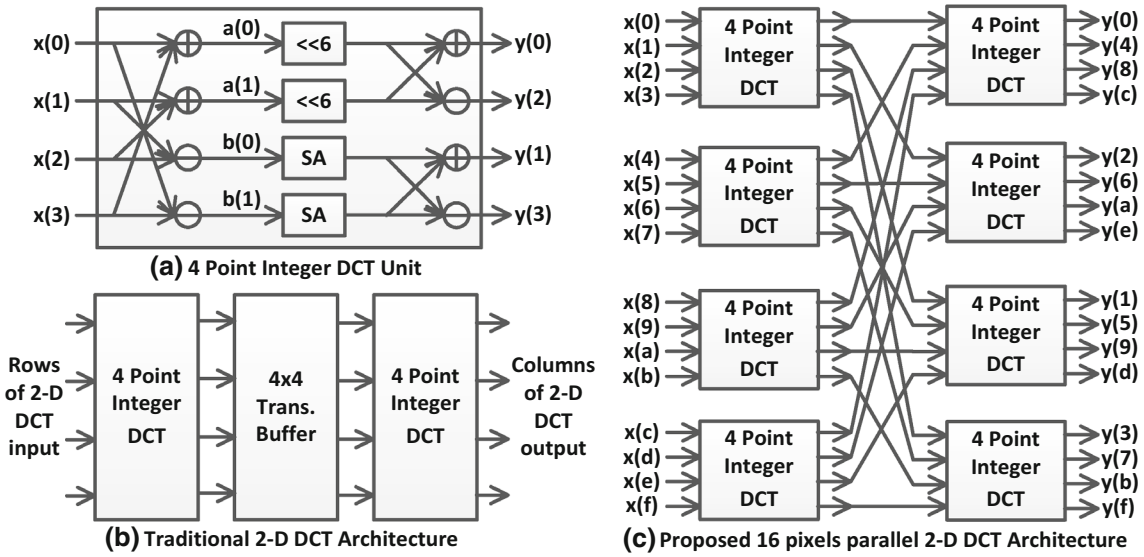


Fig. 9 Proposed 2-D DCT architecture in RDO-4. **a** The 4 point integer DCT unit. **b** Traditional 2-D DCT architecture. **c** The proposed 2-D DCT architecture

width, 20 length). The SRAM of prediction (source) buffer 1 are both (128b-width, 80 length).

- Quantized Coefficient Buffer 1. The buffer is used to store the quantized coefficients of the best prediction mode. The SRAM of the Quantized Coefficient Buffer 1 is (256b-width, 144 length).

It should be noted that there are PING-PONG buffers between the neighboring pipeline stages in RDO-8/16 module [25].

4.2 RDO mode decision pipeline scheduling

The pipeline scheduling of RDO mode decision is shown in Fig. 10. In the pipeline scheduling of RDO-4 module, 32 cycles are needed to process a 4 × 4 block with 8 prediction modes (from mode A to H). The 4 × 4 blocks in RDO-4 module are sequentially processed because of the data dependency. 128 cycles are needed to process four 4 × 4 blocks. In addition, 16 cycles are consumed to decide the best PU/CU structure. In the pipeline scheduling of RDO-8/16 module, 8 × 8 block is firstly executed, then luminance 16 × 16/32 × 32 blocks or Chroma blocks are followed. The next 8 × 8 block will start at cycle 144. In Fig. 10, the CU (luminance 16 × 16/32 × 32) insertion and Chroma insertion are proposed to improve the throughput.

The CU reordering (insertion) scheme is proposed in RDO-8/16 module. There is no data dependency for CU at different depth levels. The CU reordering scheme is proposed based on this observation. An example of CU reordering is shown in Fig. 11. CU 16 × 16 (block 4) and four 8 × 8 sub-blocks (from block 0 to 3) are used for illustration. There is no data dependency between block 0 and block 4. The two prediction modes (A, B) of block 4 can be inserted into the bubble cycles between block 0 and block 1. There are four prediction modes for 16 × 16 block as shown in Table 8. Prediction modes (C, D) of block 4 can be inserted into the bubble cycles between block 1 and block 2.

The processing order in RDO-8/16 module is in detail illustrated in Fig. 12. Blocks of CU 8 × 8, CU 16 × 16, and CU 32 × 32 are decided in CU-8/16 module to get the best CU/PU partition. The blocks of CU 16 × 16 and CU 32 × 32 are inserted into the bubble cycles during the processing of CU 8 × 8. For CU 16 × 16, the four prediction modes are separated into two phases to be decided, two prediction modes in the first phase (one block from 2, 10, 18, and 26) and two prediction modes in the second phase (one block from 4, 12, 20, and 28). For CU 32 × 32, PMI RDO mode decision is proposed in Sect. 3. For 6, 8, 14, and 16 blocks, two prediction modes are entering into RDO-8/16 module. And there is only one prediction mode for the block 22, 24, 30, and 32 to generate the standard complying quantized coefficients.

Besides the CU reordering scheme, the Chroma reordering scheme is proposed in our architecture. As illustrated in Fig. 7, the CU 32 × 32 level pipeline architecture is proposed to make the Chroma derived mode become obtainable. The Chroma block will be inserted into the bubble cycles in the RDO-4 and RDO-8/16 to improve the throughput. In RDO-8/16, the Chroma block can be inserted after 22, 24, 30, and 32 blocks as in Fig. 12. By the proposed CU reordering and Chroma reordering scheduling, the pipeline efficiency will be improved by 31.6 %.

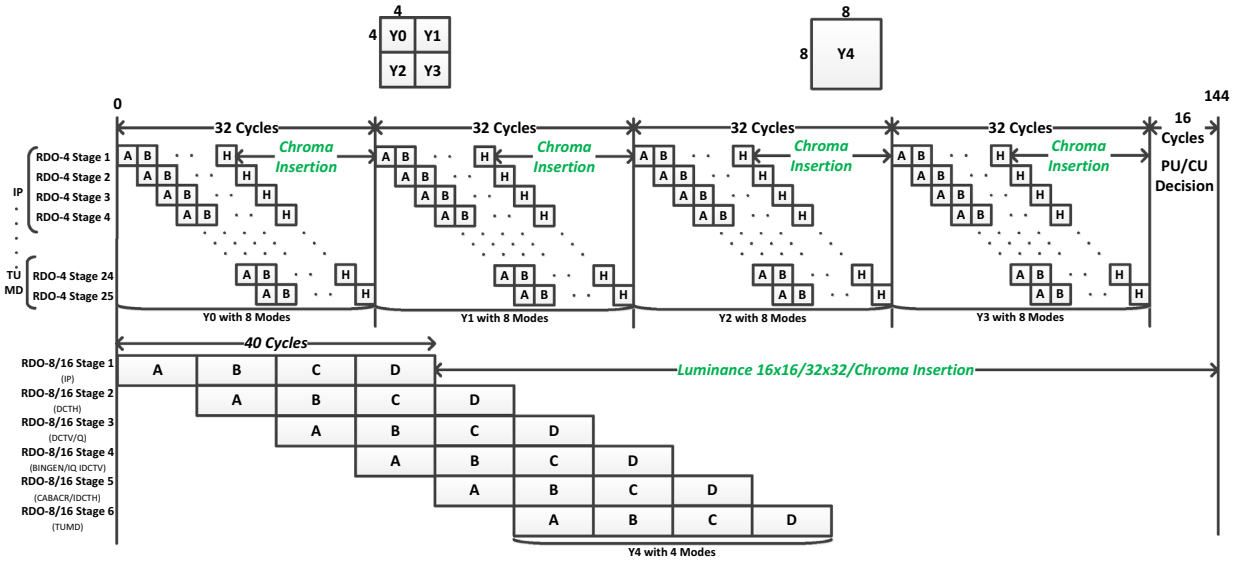


Fig. 10 RDO mode decision pipeline scheduling

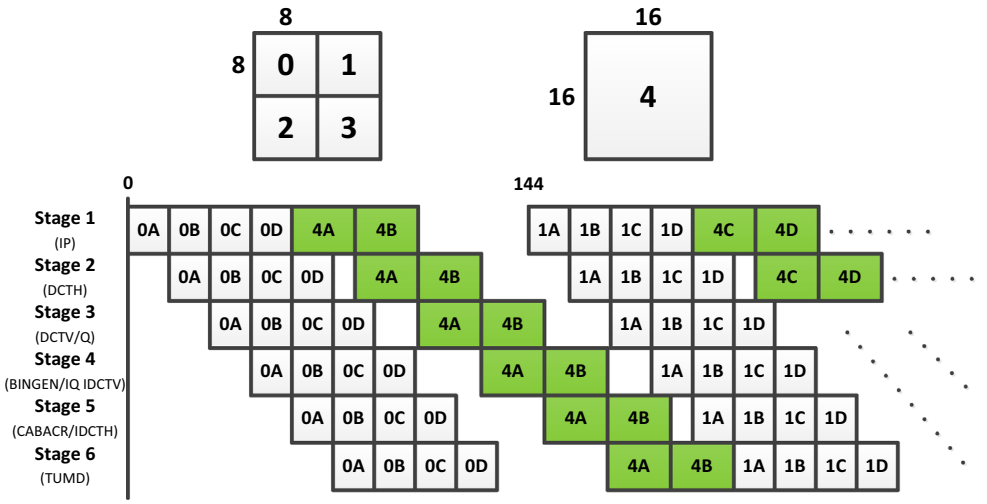


Fig. 11 An example of CU reordering scheme

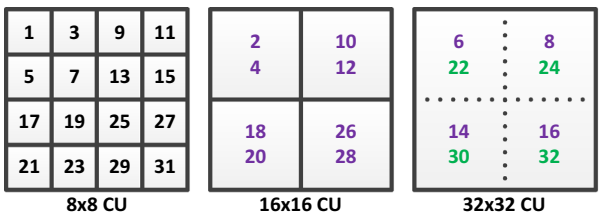


Fig. 12 CU Processing Order in RDO-8/16 module

After the pipeline scheduling, about 9616 cycles are needed to process a CTU. The needed operation *frequency* of RDO mode decision being real-time processing is given by:

$$\text{frequency} = \text{CTUCycle} \times \text{CTUnum} \times fr \tag{6}$$

In (6), the *CTUCycle* and *CTUnum* represent the cycles to process one CTU and the number of the total CTUs in one frame, respectively. *fr* is the encoding frame rate. In our architecture, *CTUCycle* is 9616. For 1080p@60fps real time processing, the *CTUnum* is 510 and the *fr* is 60. Substituting the corresponding values into (6), the 294 MHz frequency is needed to achieve the 1080p@60fps real-time processing.

4.3 Reference pixels buffer management for intra-prediction

The reference pixels buffer management for intra-prediction in RDO-8/16 is shown as in Fig. 13. The buffers are all

implemented with registers to improve the throughput. For CTU top row register buffer, the buffer size is (3 × 64 + 16) Bytes. The term of 3 × 64 is because of the supported CU 8 × 8/16 × 16/32 × 32 and the reordering scheduling. The 16 bytes is used to store the reference pixels of the above-right CTU. There are (64 + 16 + 4) bytes for CTU top left register buffer and (3 × 64) bytes for CTU left column register buffer, respectively.

4.4 CABAC rate estimation hardware architecture

The CABACR engine is shown as in Fig. 14, which can support 2/1/0 bins bit rate estimation and context state update. Three-stage pipeline architecture is implemented. {*bin0*, *bin1*} are the two bins with the same context, and {*vld0*, *vld1*} are the valid signals of the two bins. *StateIn* is the input state of the context. {*stateOut*} is the updated state. {*R0*, *R1*} are the bit-rate of the *bin0* and *bin1*, respectively. The first stage is to select the context model from the input state *stateIn* or *feedback state*. In the second stage, the 2-bin context update and the bit-rate of the *bin0* are processed in this stage. In the third stage, the bit-rate of the *bin1* is calculated. It is noted that the modules in blue color can support 1-bin bit-rate estimation and CABAC context update.

The M module in Fig. 15 is the multiplexer. The FF module in Fig. 15 is the Flip–Flop register. The implementation of the *S, D* modules is shown as in Table 9. The *S* module is the implementation of state transition table. And the *D* performs a updating for two successive bins [31]. The *R* module is the implementation of the table-based bit estimate [23].

The block diagram of the CABACR module is shown in Fig. 15. The CABAC engines are parallelized to achieve the parallelism of 16 coefficients processing. The bit-rate *R* is the sum of all the coded bins. The updated context will output to the TUMD and CUMD module. After the mode decision, the best context state will be output to the context state buffer CS0/1. Utilizing the proposed architecture, the context will be updated in real time.

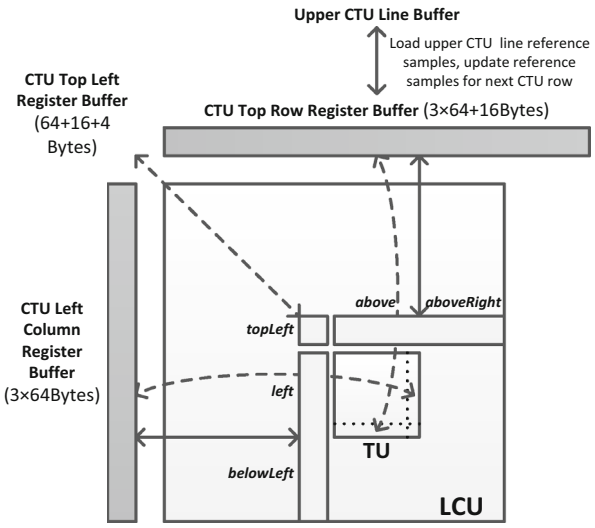


Fig. 13 Reference Pixels Buffer Management in RDO-8/16

5 Implementation results

5.1 Coding performance comparison

Table 10 shows the coding performance comparison of the proposed intra-mode decision and the state-of-the-art work [17, 18]. The setting of the reference software is illustrated as in Sect. 3. The performance gain or loss is measured

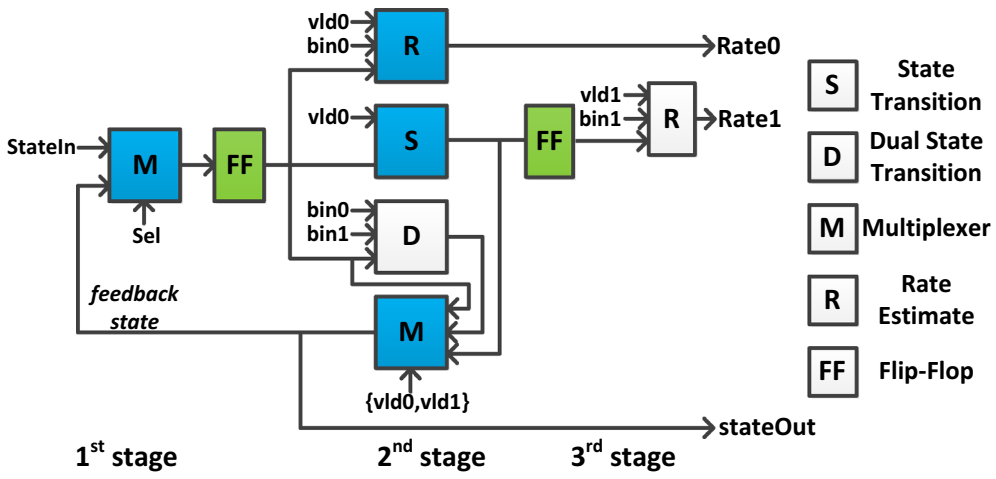


Fig. 14 2/1/0 Bins CABACR engine

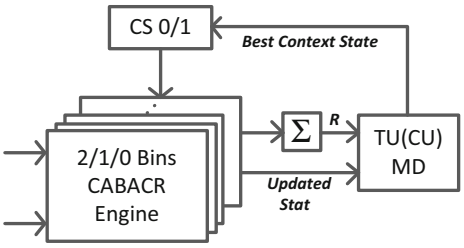


Fig. 15 CABACR architecture block diagram

Table 9 (a) S Modules implementation, (b) D Modules Implementation (State Transition Table)

(a)				
pStateIdx	0	1	2	…
transIdxLps	0	0	1	…
transIdxMps	1	1	3	…
(b)				
pStateIdx	0	1	2	…
transIdxLL	0	0	0	…
transIdxLM	1	1	2	…
transIdxML	0	1	2	…
transIdxMM	2	3	4	…

with respect to HM-13.0. BDR (R) and encoding time reduction (T) are used to estimate the average bit-rate difference and computation complexity reduction.

From the experimental results in Table 10, it is observed that the proposed intra-mode decision scheme achieves 41.6 % computational complexity reduction with 4.3 % BDR increase on average. The coding performance degradation shows significant for *Kimono* and *Johnny*. It is partially due to the disabling coding tool of TU 32 × 32. With disabling coding tool of TU 32 × 32, the BDR is increased by 6.1 and 3.4 % for *Kimono* and *Johnny*, respectively.

Our intra-mode decision shows comparable coding efficiency compared with work [18]. In [18], two CU depth levels will enter into RDO mode decision to decide the best CU structure, which lead to more complexity reduction. However, our work can achieve smaller chip area and higher throughput than [18], which will be further illustrated in Sect. 4.2. In [17], 8 × 8 CUs and 4 × 4 PUs are eliminated for high-resolution applications. For the test sequences, the coding performance degradation is significant, with 15.7 % BDR increase on average.

5.2 Hardware implementation results

The proposed RDO mode decision VLSI architecture is implemented with Verilog-HDL language and synthesized by Design Compiler with SMIC 55 nm 1P8 M standard

Table 10 Coding performance comparison (vs. HM-13.0)

Class	Sequence	[18]		[17]		Proposed	
		R (%)	T (%)	R (%)	T (%)	R (%)	T (%)
A	*Traffic*	4.9	62.5	13.9	74.7	4.0	33.3
	PeopleOnstreet	5.1	63.1	17.0	76.5	3.1	36.4
B	*BasketballDrive*	7.1	62.3	14.8	76.8	7.7	37.8
	BQTerrace	4.8	65.2	12.4	78.4	3.1	37.4
	Cactus	4.8	73.4	14.1	74.3	4.1	37.6
	Kimono	4.6	69.2	7.9	75.6	**11.2**	38.1
	ParkScene	3.9	59.7	9.6	74.8	3.2	37.3
C	*BasketballDrill*	4.7	61.5	19.3	84.7	4.2	44.3
	BQMall	4.5	59.6	15.3	83.3	3.8	45.2
	PartyScene	3.5	58.3	14.6	82.5	1.9	43.3
	RaceHorses	3.7	57.4	11.8	84.4	2.8	44.5
D	*BasketballPass*	5.1	59.6	15.7	83.5	4.6	44.4
	BlowingBubbles	3.7	55.3	11.5	84.7	2.2	42.1
	BQSquare	2.5	57.2	16.7	83.5	1.3	46.3
	RaceHorses	3.6	59.3	13.9	82.3	3.2	43.3
E	*FourPeople*	6.2	62.6	18.4	78.5	4.1	43.4
	Johnny	6.7	61.7	18.8	76.8	**9.8**	42.5
	KristenAndSara	6.2	64.7	19.8	78.2	6.3	44.2
F	*ChinaSpeed*	3.4	59.8	19.4	76.3	2.7	45.6
	SlideEditing	3.2	62.9	21.7	78.4	2.0	43.2
	SlideShow	6.5	61.4	23.2	79.7	6.4	42.5
Average		4.7	61.8	15.7	79.4	4.3	41.6

CMOS technology under a timing constraint of 294 MHz. Table 11 lists the gate count of the proposed intra-mode decision architecture. The proposed intra-mode decision architecture is implemented with 1571.7 K gate count, which includes the gate count of on-chip SRAMs.

The comparisons with the state of the art are shown in Table 12. The maximum throughput in Table 12 is calculated as (7). In (7), *picWidth* is the picture width, *picHeight* is the picture height, *fr* is the encoding frame rate.

$$\text{Throughput} = \text{picWidth} \times \text{picHeight} \times fr \tag{7}$$

This paper can achieve 1080p@60fps real-time processing at the hardware working frequency of 294 MHz. The intra-quality (BDR) of the proposed scheme is increased by 4.3 % compared to the HM-13.0. The power consumption of the work is 194 mW.

Compared to work [18], our proposed architecture can achieve both the chip area reduction and the throughput increase. The chip area reduction comes from the disabling coding tool of TU 32 × 32 and lower pixel parallelism. In [18], the parallelism of 128 pixels is adopted for intra-predictor, while 16 pixels parallelism is adopted in our work. The throughput increase of the proposed work comes from the fast algorithms, the CTU level pipelined RMD and RDO mode decision, the parallel RDO mode decision VLSI architecture and the efficient CU and Chroma reordering scheduling schemes. Although the complexity is decreased by CU/PU pre-decision in [18], the sequential processing of the rough search, fine search and reconstruction for PU mode decision causes the throughput degradation.

Compared to work [17], our HEVC intra-encoder achieves better intra-coding performance. In [17], an 8192 × 4320@30fps real-time HEVC encoder is implemented with 8350 K gate. The elimination of the complex 8 × 8 CU and 4 × 4 PU, and fully parallel (64 × 64 intra-CU, 32 × 32 intra-CU and 16 × 16 intra-CU are parallelized) hardware architecture are adopted to meet the high throughput requirement. These cause the coding performance degradation and chip area increase, simultaneously.

6 Conclusion

In this paper, fast algorithms and VLSI architecture for HEVC intra-mode decision are proposed. The fast algorithms are proposed to remove the data dependency and

Table 11 List of gate count

Module	Sub-Module	Gate Count (K) (NAND2, including SRAM)
RMD	–	**325.8**
RDO-4	–	**282.5**
RDO-8/16	IP	87.3
	DCTH	86.5
	DCTV/Q	97.4
	IQ/IDCTV	110.3
	IDCTH	96.7
	BINGEN	56.8
	CABACR	120.4
	TUMD	42.2
	Total	**697.6**
RDO-Top	–	**265.8**
Total.	–	**1571.7**

Table 12 Comparisons between this paper and the State-of-the-art

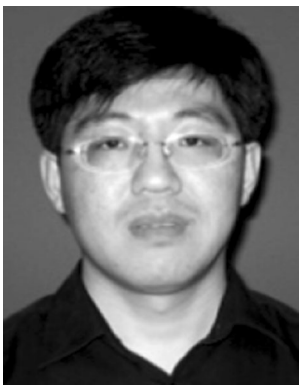
	Proposed	[18]	[17]
Standard	HEVC	HEVC	HEVC
Coding tools	Intra	Intra	Intra & Inter
Resolution	1920 × 1080 @60fps	1920 × 1080 @44fps	8192 × 4320 @30fps
Max.Throughput (Mpixels/s)	124	91	1062
Intra-Quality (vs. HM) (%)	4.3	4.7	15.7
Technology (nm)	55	90	28
Gate Count (K)	1571.7	2269	8350
Frequency (MHz)	294	357	312
Power (mW)	194	218	708

reduce computational complexity, which contain the fast RMD, the PMI RDO mode decision, the parallelized context adaption and the Chroma-free CU/PU decision. The parallelized VLSI architecture and the efficient pipeline scheduling are proposed for HEVC intra-mode decision. The CU reordering and Chroma reordering scheduling techniques are proposed to improve the throughput. The proposed intra-mode decision achieves 41.6 % complexity reduction with 4.3 % BDR increase on average. The intra-mode decision design is implemented with 1571.7 K gate count in 55 nm CMOS technology. The implementation results show that our design can achieve the 1080p@60fps real time processing at 294 MHz operation frequency.

References

1. Ohm, J.-R., Sullivan, G.J., Schwarz, H., Tan, T.K., Wiegand, T.: Comparison of the coding efficiency of video coding standards-including high efficiency video coding (HEVC). IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1668–1683 (2012)
2. Sullivan, G.-J., Ohm, J.-R., Han, W.-J., Wiegand, T.: Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1649–1668 (2012)
3. Lainema, J., Bossen, F., Han, W.-J., Min, J., Ugur, K.: Intra coding of the HEVC standard. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1792–1801 (2012)
4. Nguyen, T., Marpe, D.: Performance analysis of HEVC-based intra coding for still image compression. In: Picture Coding Symposium (PCS), pp. 233–236 (2012)
5. Piao, Y., Min, J., Chen, J.: Encoder improvement of unified intra prediction. JCTVC-C207, Guangzhou (2010)
6. Jiang, W, Ma, H., Chen, Y.: Gradient based fast mode decision algorithm for intra prediction in HEVC. In International Conference on Consumer Electronics, Communications and Networks (CECNet) (2012)
7. Pan, F., Lin, X., Rahardja, S., Lim, K., Li, Z., Wu, D., Wu, S.: Fast mode decision algorithm for intra prediction in H.264/AVC video coding. IEEE Trans. Circuits Syst. Video Technol. **15**(7), 813–822 (2005)
8. Zhang, H., Ma, Z.: Fast intra mode decision for high efficiency video coding (HEVC). IEEE Trans. Circuits Syst. Video Technol. **24**(4), 660–668 (2014)
9. Shen, L., Zhang, Z., An, P.: Fast CU size decision and mode decision algorithm for HEVC intra coding. IEEE Trans. Consumer Electronics **59**(1), 207–213 (2013)
10. Nishikori, T., Nakamura, T., Yoshitome, T., Mishiba, K.: A fast CU decision using image variance in HEVC intra coding. In Proceedings of IEEE ISIEA, pp. 52–56 (2013)
11. Huang, H., Zhao, Y., Lin, C., Bai, H.: Fast bottom-up pruning for HEVC intraframe coding. In Proceedings of Visual Communications and Image Processing (VCIP), pp. 1–5 (2013)
12. Huang, C., Tikekar, M., Chandrakasan, A.P.: Memory-hierarchical and mode-adaptive HEVC intra prediction architecture for quad full HD video decoding. IEEE Trans. Very Large Scale Integr. VLSI Syst. **2**(7), 1515–1525 (2014)
13. Liu, Z., Wang, D., Zhu, H., Huang, X.: 41.7BN-Pixels/s reconfigurable intra prediction architecture for HEVC 2560 × 1600 Encoder. In: Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2634–2638 (2013)
14. Meher, P.K., Park, S.Y.: Efficient integer DCT architectures for HEVC. IEEE Trans. Circuits Syst. Video Technol. **24**(1), 168–178 (2014)
15. Zhu, J., Liu, Z., Wang, D.: Fully Pipelined DCT/IDCT/Hadamard unified transform architecture for HEVC Codec. In: Proceedings of IEEE International Symposium of Circuits System (ISCAS), pp. 677–680 (2013)
16. Li, F., Shi, G.: A pipelined architecture for 4 × 4 intra frame mode decision in the high efficiency video coding. In Multimedia Signal Processing (MMSP), pp. 1–5 (2011)
17. Tsai, S.-F., Li, C.-T, Chen, H.-H., Tsung P.-K., Chen, K.-Y., Chen, L.-G.: A 1062Mpixels/s 8192x4320p high efficiency video coding (H.265) Encoder Chip. In Symposium on VLSI Circuits (VLSIC), pp. C188–C189 (2013)
18. Zhu, J., Liu, Z., Wang, D., Han, Q., Song, Y.: HDTV1080p HEVC Intra encoder with source texture based CU/PU mode predecision. In: Design Automation Conference (ASP-DAC), pp. 367–372, 20–23 (2014)
19. Ding, L.-F., Chen, W.-Y., Tsung, P.-K., Chuang, T.-D., Hsiao, P.-H., Chen, Y.-H., Chiu, H.-K., Chien, S.-Y., Chen, L.-G.: A 212 MPixels/s 4096 × 2160p multiview video encoder chip for 3D/quad full HDTV applications. IEEE J. Solid-State Circuits **45**(1), 46–58 (2010)
20. Mochizuki, S., Shibayama, T., Hase, M., Izuhara, F., Akie, K., Nobori, M., Imaoka, R., Ueda, H., Ishikawa, K., Watanabe, H.: A 64 mW high picture quality H.264/MPEG-4 video codec IP for HD mobile applications in 90 nm CMOS. IEEE J. Solid-State Circuits **43**(11), 2354–2362 (2008)
21. He, G., Zhou, D., Fei, W., Chen, Z., Zhou, J., Goto, S.: High-performance H.264/AVC intra-prediction architecture for ultra high definition video applications. IEEE Trans. Very Large Scale Integr. VLSI Syst. **22**(1), 76–89 (2014)
22. Kuo, H.-C., Wu, L.-C., Huang, H.-T., Hsu, S.-T., Lin, Y.-L.: A low-power high-performance H.264/AVC intra-frame encoder for 1080pHD video. IEEE Trans. Very Large Scale Integr. VLSI Syst. **19**(6), 925–938 (2011)
23. Bossen, F.: CE1: Table-based bit estimation for CABAC, JCTVC-G763, 7th Joint Collaborative Team on Video Coding (JCTVC) Meeting, Geneva, Switzerland (2011)
24. Zhao, L., Zhang, L., Ma, S., Zhao, D.: Fast mode decision algorithm for intra prediction in HEVC. In: IEEE Visual Communications and Image Processing (VCIP), pp. 1–4 (2011)
25. Zhu, C., Jia, H., Zhang, S., Huang, X., Xie, X., Gao, W.: On a highly efficient RDO-based mode decision pipeline design for AVS. IEEE Trans. Multimedia **15**(8), 1815–1829 (2013)
26. Shang, Q., Fan, Y., Shen, W., Shen, S., Zeng, X.: Single-port SRAM-based transpose memory with diagonal data mapping for large size 2-D DCT/IDCT. IEEE Trans. Very Large Scale Integr. VLSI Syst. **22**(11), 2422–2426 (2014)
27. Bjontegaard, G.: Calculation of average PSNR difference between RD-curves. In: 13th VCEG-M33 Meeting, Austin, TX (2001)
28. ITU-T, ISO/IEC JTC 1.: *High Efficiency Video Coding*, ITU-T Rec. H.265, ISO/IEC 23008-2 (2014)
29. Sole, J., Joshi, R., Nguyen, N., Ji, T., Karczewicz, M., Clare, G., Henry, F., Duenas, A.: Transform coefficient coding in HEVC. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1765–1777 (2012)
30. Zhu, C., Jia, H., Liu, J., Ji, X., Lv, H., Xie, X., Gao, W.: Multi-level low-complexity coefficient discarding scheme for video encoder. In: Proceedings of IEEE International Symposium of Circuits System (ISCAS), pp. 5–8 (2014)
31. Zhou, J., Zhou, D., Fei, W., Goto, S.: A High-performance CABAC encoder architecture for HEVC and H.264/AVC. In IEEE International Conference on Image Processing (ICIP), pp. 1568–1572 (2013)

Xiaofeng Huang received the B.S. degree in Microelectronics from The Nanjing University of Posts and Telecommunications, Nanjing, China, in 2010. He is currently pursuing the Ph.D. degree in the School of Electronics Engineering and Computer Science, Peking University, Beijing, China. His research interests include video coding, VLSI design, digital IC design and digital signal processing.

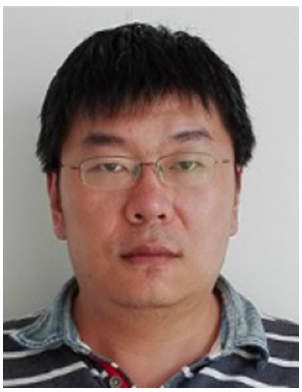


Huizhu Jia received his Ph.D. degree in electrical engineering from the Chinese Academy of Sciences, Beijing, China, in 2007. He is currently with the National Engineering Laboratory for Video Technology, Peking University, Beijing. His research interests include: ASIC design, image processing, multimedia data compression and VR.

Binbin Cai received the B.S. degree in electrical engineering and automation from The Institute of Disaster Prevention Science and Technology, Langfang, China, in 2012. He is currently pursuing the master degree in the School of Software and Microelectronics, Peking University, Beijing, China. His research interests include video encoding, image processing and VLSI chip design.

Chuang Zhu received the B.S. degree in computer science from The North University of China, Taiyuan, China, in 2008. He is currently pursuing the Ph.D. degree in the School of Electronics Engineering and Computer Science, Peking University, Beijing, China. His research interests include video encoding, image processing and VLSI chip design.

Jie Liu received the B.S. degree in in Electronic Information and Engineering from An yang Institute of Technology, China, in 2011, and the M.S. degree in Integrated Circuits and Systems from Peking University, Beijing, China, in 2014. He is currently pursuing the Ph.D. degree in the School of Electronics Engineering and Computer Science, Peking University, Beijing, China. His research interests include VLSI design for Video coding with H264, AVS and HEVC standards and image processing.



Mingyuan Yang received the Ph.D. degree in electronic and electrical engineering from Loughborough University, UK. He was a research associate in university of central Lancashire, UK. Now he is a senior research engineer in Boyahualu technology, China. He has published more than 20 conference and journal papers, 10 standard proposals and 30 patents in image video areas. His main research interests include image and video coding, video streaming and transmission.

Don Xie received his Ph.D. degree in Electrical Engineering, University of Rochester, USA. He was a Senior Scientist at Eastman Kodak Company, New York, USA, from 1994 to 1997; a Principal Scientist at Broadcom Corporation, California, USA, from 1997 to 2009. He is currently a Professor with the Department of Electrical Engineering, Peking University, Beijing, China. His research interests include multimedia SoC design, embedded system. He holds 24 U.S. Patents.

Wen Gao (M'92-SM'05-F'09) received the Ph.D. degree in electronics engineering from the University of Tokyo, Japan, in 1991. He is a professor in computer science at Peking University, China. Before joining Peking University, he was a professor at Harbin Institute of Technology from 1991 to 1995, and a professor at the Institute of Computing Technology of Chinese Academy of Sciences from 1996 to 2006. He has published extensively including five books and over 600 technical articles in refereed journals and conference proceedings in the areas of image processing, video coding and communication, pattern recognition, multimedia information retrieval, multimodal interface, and bioinformatics. Prof. Gao served or serves on the editorial board for several journals, such as IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Multimedia, IEEE Transactions on Autonomous Mental Development, EURASIP Journal of Image Communications, Journal of Visual Communication and Image Representation. He chaired a number of prestigious international conferences on multimedia and video signal processing, such as IEEE ICME and ACM Multimedia, and also served on the advisory and technical committees of numerous professional organizations. He is a member of Chinese Academy of Engineering.