A Dual-Clock VLSI Design of H.265 Sample Adaptive Offset Estimation for 8k Ultra-HD TV Encoding

Jianbin Zhou, Dajiang Zhou, *Member, IEEE,* Shihao Wang, Shuping Zhang, Takeshi Yoshimura, *Member, IEEE,* and Satoshi Goto, *Life Fellow, IEEE*

***Abstract*— Sample adaptive offset (SAO) is a newly introduced in-loop filtering component in H.265/High Efficiency Video Coding (HEVC). While SAO contributes to a notable coding efficiency improvement, the estimation of SAO parameters dominates the complexity of in-loop filtering in HEVC encoding. This paper presents an efficient VLSI design for SAO estimation. Our design features a dual-clock architecture that processes statistics collection (SC) and parameter decision (PD), the two main functional blocks of SAO estimation, at high- and low-speed clocks, respectively. Such a strategy reduces the overall area by 56% by addressing the heterogeneous data flows of SC and PD. To further improve the area and power efficiency, algorithm-architecture co-optimizations are applied, including a coarse range selection (CRS) and an accumulator bit width reduction (ABR). CRS shrinks the range of fine processed bands for the band offset estimation. ABR further reduces the area by narrowing the accumulators of SC. They together achieve another 25% area reduction. The proposed VLSI design is capable of processing 8k at 120-frames/s encoding. It occupies 51k logic gates, only one-third of the circuit area of the state-of-the-art implementations.**

***Index Terms*— Dual-clock architecture, H.265, high efficiency video coding (HEVC), sample adaptive offset (SAO), video coding, VLSI.**

I. Introduction

WHILE 1080p High Definition (HD) and 4k ultra-HD formats are dominating today's high-end video applications, 8k ultra-HD is being promoted as the next-generation video specification. The 8k is expected to deliver extremely high visual experience by having up to 7680 × 4320 pixels per frame and 120 frames/s [1]. To transmit such a huge data throughput (TP) in the communication channel, deep compression from the latest video coding technology, High Efficiency Video Coding (H.265/HEVC) [2], [3], plays a crucial role. The implementation of the corresponding video codecs, however, is challenged by the multiplication of the ultrahigh TP requirement and an increased complexity per pixel. Compared with the previous H.264/Advance Video Coding (AVC) standard [4], H.265/HEVC doubles coding efficiency by employing a number of new coding tools. Especially, a sample adaptive offset (SAO) component is newly introduced as one of the in-loop filters (ILFs), which contributes to up to 18% Bjøntegaard delta (BD)-rate reduction [5].

In H.264/AVC, deblocking filter (DBF) [6] is the only ILF. Its VLSI implementation has been discussed in many previous works [7]–[10]. In H.265/HEVC, DBF [11] has been simplified [12]–[17], and SAO dominates the complexity of ILF especially in a video encoder. Several previous works discussed SAO's implementation. Joo *et al.* [18], [19] proposed to utilize the intraprediction mode to predict the edge offset (EO) type, so that the number of EO types could be reduced to save the encoding time. Choi and Joo [20] evaluated several algorithm-level improvements for SAO. Chen *et al.* [21] developed a low complexity SAO algorithm based on class combination, band offset (BO) predecision, and merge separation category. Rediess *et al.* [22], [23] discussed the architectures of statistics collection (SC) and parameter decision (PD), the two main components of SAO. Mody *et al.* [24] designed an SAO estimation architecture supporting 4k at 60-frames/s encoding. Zhu *et al.* [25], [26] developed a fast SAO estimation algorithm and its VLSI architecture supporting 8k encoding. The complete implementations of SAO estimation [24]–[26] both require relatively large circuit area, which still has plenty room for improvement.

This paper aims at designing an efficient VLSI architecture of SAO estimation in H.265/HEVC. To achieve high area efficiency, we propose three techniques.

1) *Dual-Clock SAO Architecture:* The highly heterogeneous data flow of SC and PD in SAO causes each part to require a completely different preference in working frequency. Such a different preference is the main obstacle for an efficient implementation. This technique addresses heterogeneous data flow by separately driving SC and PD at high- and low-speed clocks, respectively, so that each part could be integrated together efficiently. It reduces the overall area by 56%, from 156k to 68k gates.
2) *Coarse Range Selection (CRS) for BO:* Based on the analysis of band distributions in each coding tree block (CTB), and on hardware resources for finding best

Manuscript received February 24, 2016; revised June 4, 2016; accepted July 13, 2016. This work is supported by Waseda University Graduate Program for Embodiment Informatics under Grant FY2013-FY2019. *(Corresponding author: Dajiang Zhou.)*

The authors are with the Graduate School of Information, Production and Systems, Waseda University, Kitakyushu 808-0135, Japan (e-mail: khshu@fuji.waseda.jp; zhou@fuji.waseda.jp; wshh1216@moegi.waseda.jp; zhangshuping@fuji.waseda.jp; t-yoshimura@waseda.jp; goto@waseda.jp).

Digital Object Identifier 10.1109/TVLSI.2016.2593581

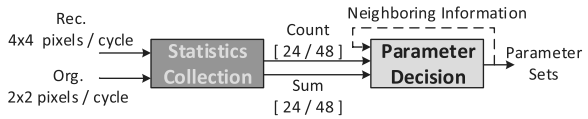


Fig. 1. Overview of SAO. Details of the proposed SC engine and PD engine are in Figs. 6 and 7, respectively.

bands, this technique estimates the range of bands before SC with an accuracy of 60%–80% and shrinks the range of fine processed bands 32 to 8 to reduce the circuit area.

3) *Accumulator Bit Width Reduction (ABR):* By exploiting the mutual exclusion relationship among categories/bands existed in the accumulation process in SC, this technique carries out an early termination to accumulators reaching a threshold, to further reduce their circuit area.

Our VLSI implementation employing the above techniques occupies 51k logic gates, which is only one-third of circuit size of [25], at the same TP and comparable coding efficiency. With a high-speed clock of 1.3 GHz and a low-speed clock of 217 MHz, 8k at 120-frames/s SAO real-time encoding can be achieved.

The rest of this paper is organized as follows. Section II gives an introduction to SAO in H.265, its data flow, and several hardware friendly approaches for the design. Section III analyzes the data flow of SAO, the key challenge, and introduces the first proposed technique. Section IV-A analyzes the characteristics of BO and introduces our second proposed technique utilizing those characteristics. Section IV-B explains how mutual exclusion among categories/bands generates inefficiency in hardware utilization, and gives our third proposed technique. Section V shows the implementing results and gives some analysis on the performance of the proposed design, followed by the conclusion in Section VI.

II. SAO Algorithm

SAO aims at reducing the distortion of the reconstructed pictures, by adaptively adding offsets to the reconstructed samples at both encoder and decoder. The SAO parameters, i.e., how the offsets should be generated and applied, are signaled at the coding tree unit (CTU)[1] level. The offset to be applied depends on the classification of the target sample. There are two kinds of classifiers: EO and BO. The sample classification of EO depends on the comparison between the current sample and its neighboring ones, while the sample classification of BO depends on the value of current sample itself. The optimal classifier and the offsets for each CTU are found during the encoding process, called SAO estimation, which comprises the SC and PD phases, as shown in Fig. 1. In SC, the BO and EO classifiers classify each reconstructed sample in a CTU into different bands and categories, respectively. The classification statistics of the current CTU are collected. In PD, based on the statistics and the neighboring (left and upper) SAO parameters, the optimal parameter sets achieving the lowest rate-distortion cost are found. The parameter sets include the SAO mode, types, and offsets, as shown in Table I.

[1]In HEVC, a picture is partitioned into CTUs scanned in a raster order. A CTU consists of a luma CTB and the corresponding chroma CTBs and syntax elements.

TABLE I

Output (Parameter Sets) of SAO. Luma and Chroma Share the Same Set of Parameters in Gray Parts. Cb and Cr Always Share the Same Mode and Type

SAO Mode	SAO Type	SAO Offset	Type AuxInfo
OFF :0	N/A	N/A	N/A
New_Mode :1	EO_0 :0	Offset [0:3]	
	EO_90 :1		
	EO_135 :2		
	EO_45 :3		
	BO :4		0-28
Merge :2	Merge_Upper :0	Follow merged CTU.	
	Merge_Left :1		

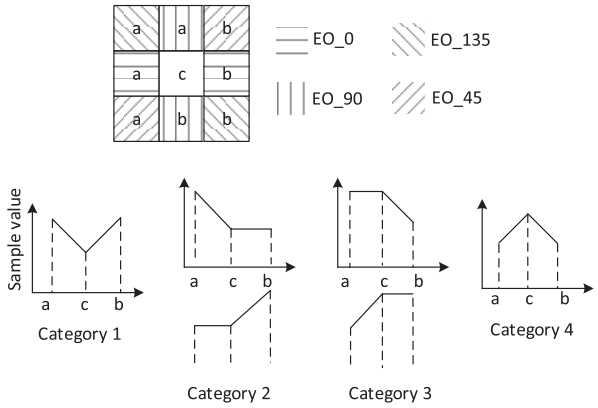


Fig. 2. EO patterns and categories.

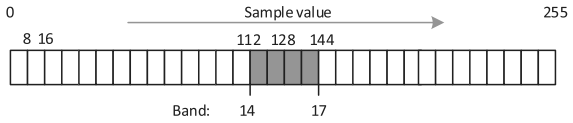


Fig. 3. Value of an 8-bit sample [dynamic range $(0–2^{\text{BitDepth}-1})$] is evenly divided into 32 bands. The best consecutive four bands are chosen as candidates, e.g., Bands 14–17.

A. Statistics Collection

For EO, the category of each sample is decided according to its relationship with neighboring samples, following four patterns, the horizontal (EO_0), the vertical (EO_90), and two diagonal (EO_45 and EO_135) directions, as shown in Fig. 2. A sample that falls into none of these categories is classified into category 0.

For BO, the band of a sample is decided according to the value range it falls in. The entire dynamic range $(0–2^{\text{BitDepth}-1})$ is evenly divided into 32 bands, as shown in Fig. 3. An 8-bit sample is classified into band K if it ranges from 8k to 8k + 7. Based on the statistics collected within a CTB, the best consecutive four bands and their corresponding offsets are chosen as candidates to compared with the EO patterns.

During SC, each original and reconstructed sample is scanned within a CTB. Based on the result of classification on reconstructed samples, differences between original and

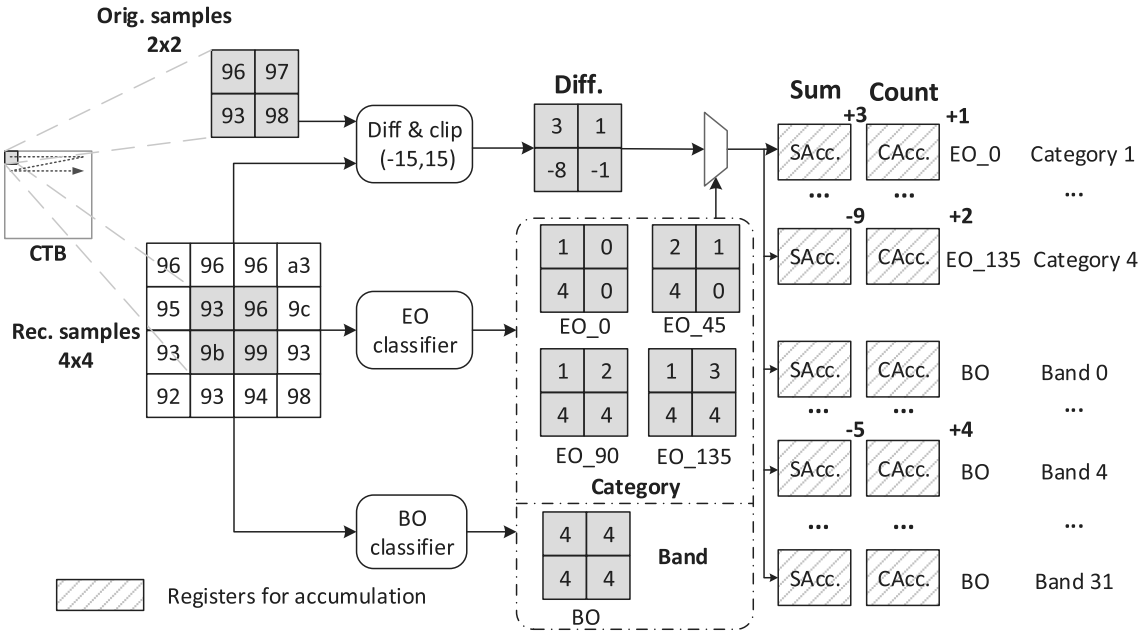


Fig. 4. Example to illustrate the SC process. SAcc and CAcc are the abbreviations for accumulators of Sum and Count, e.g., BO classification is performed to the 2×2 reconstruction samples (0×93, 0×96, $0 \times 9b$, and 0×99). Since all these samples belong to Band 4, the differences (Org.-Rec.) belong to Band 4 are summed up. The SAcc and the CAcc of Band 4 add to -5 and 4, respectively.

TABLE II
COMPARISON OF MODIFICATIONS MADE ON THE HM-16.0

No.	Name	HM-16.0	ICIP14 [25]	Proposed
1	Number of iteration for offsets	At most 7 iteration	No iteration	
2	Evaluation method for best band	RD-Cost	Distortion	
3	Rate	CABAC	constant probability model[25]	
4	Normalization of RD-cost	The RD-cost of New RDO mode	The RD-cost of Merge mode	
5	The range of difference	[-1023,1023]	[-7,7]	[-15,15]
6	Samples unused in SC (Fig. 5)	Region A	Region A∪B	Region A∪C
7	Number of bands	32	32	8
8	Accumulator bit width reduction	-	-	Accumulator terminates when reaching a threshold

reconstructed samples are accumulated to the corresponding category in each EO pattern as well as the corresponding band in BO. After an entire CTB is processed, the sum of the differences (Sum) and the number of occurrences (Count) of that category/band are output to PD for calculating the offsets and distortion. Fig. 4 shows an example about how it proceeds in our design.

B. Parameter Decision

The purpose of PD is to decide the parameters for the current CTU, based on the statistics collected in SC. The parameters include the SAO mode, SAO type, auxiliary information, and four offsets. The possible outputs are listed in Table I. The set of parameters with lowest rate-distortion (RD) cost is chosen as the one to be coded. RD cost is defined as

$$\text{Cost} = D + \lambda * \text{rate} \tag{1}$$

where rate is the number of bits to code the parameters, and λ is the Lagrange multiplier. Distortion between the original and reconstructed samples modified by SAO can be described by the following equation:

$$\begin{aligned} D_{\text{post}} &= \sum_{c \in \text{CTB}} (\text{org}(c) - \text{rec}_{\text{post}}(c))^2 \\ &= \sum_{c \in \text{CTB}} (\text{org}(c) - (\text{rec}_{\text{pre}}(c) + \text{offset}))^2 \end{aligned} \tag{2}$$

where the offset is calculated by the Sum and Count from SC

$$\text{offset} = \text{Sum}/\text{Count}. \tag{3}$$

By evaluating all the bands, EO patterns, and the merge candidates, the parameter sets with minimum cost are chosen as the final decision. The comparison of distortions can be simplified by eliminating the org and rec in D and D_{post} as the following equations. The details can be referred in [5]

$$\begin{aligned} \delta D &= D_{\text{post}} - D_{\text{pre}} \\ &= \text{Count} * \text{offset}^2 - 2 * \text{offset} * \text{Sum}. \end{aligned} \tag{4}$$

C. Hardware Friendly SAO

To improve the algorithm's friendliness to hardware implementation, a previous work [25] made modifications to the

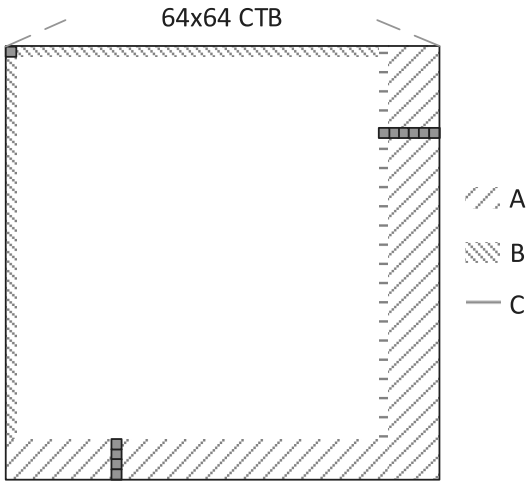


Fig. 5. Samples not used in SC, referring to No. 6 in Table II.

original SAO algorithm in the HM reference software, as listed in Table II from Mods 1 to 4.

In this paper, we further apply Mods 5–8. Mods 5 enlarges the range of difference between original samples and reconstructed samples for a higher precision. Mods 6 utilizes the top and left boundary samples for SC to increase the calculation accuracy. As Mod 7, we proposed a technique to estimate the most probably selected bands in a CTB and to reduce the searching space for the best bands from 32 to 8 bands, the details of which are explained in Section IV-A. As Mod 8, we propose a bit width reduction technique for the accumulators, with details given in Section IV-B.

III. Dual-Clock Architecture

A. Heterogeneous Data Flows of SC and PD

The main obstacle to an efficient SAO implementation comes from the highly heterogeneous data flows of SC and PD. The SC for each EO or BO classifier comprises many simple iterations. On the other hand, PD involves significantly less iterations (56 or less for each CTU) with each of them being much more complex.

The system TP can be regarded as the product of clock frequency (freq) and parallelism (N) in the number of samples processed per clock cycle

$$\text{Throughput} = \text{freq} * N. \tag{5}$$

The enhancement of TP can come from the increase of either freq or N. The serial characteristics of SC and the large number of iterations involved, however, make SC inefficient to be parallelized. As shown in the gray part in Figs. 6 and 7, the hardware components of these parts have an quadratic growth in area with the increase of N. Detailed quantitative analysis will be given in Section III-B. In the meanwhile, the function of SC decides that a short critical path can be achieved, and thus, a high frequency is preferred. However, a high working frequency is not preferred in PD, because: 1) it does not need many clock cycles to perform the limited number of iterations involved for each CTU and 2) each iteration involves

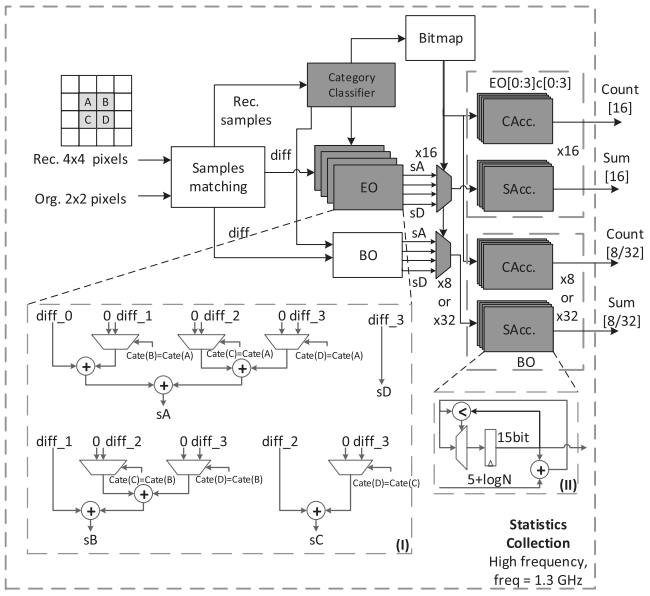


Fig. 6. Architecture of 2×2 SC engine. The details of the dark gray part in Fig. 1. SAcc and CAcc are the abbreviations for the accumulators of Sum and Count.

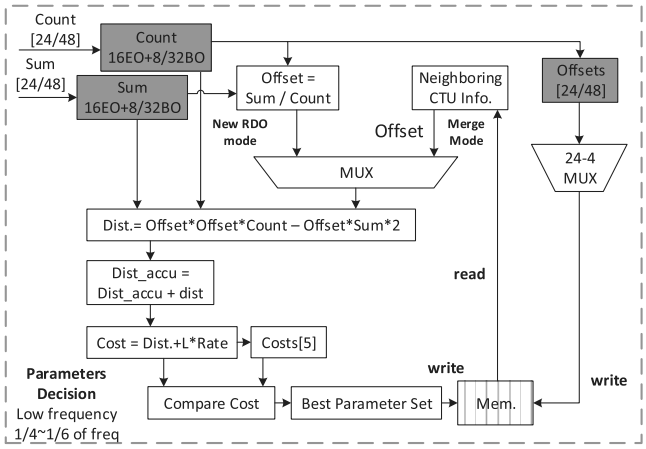


Fig. 7. Proposed architecture of PD Engine. The details of the light gray part in Fig. 1.

the complex computation that results in a long critical path. The big difference in preference to the selection of working frequencies, thus, becomes the key challenge for integrating SC and PD efficiently.

B. Optimal Clock Frequency of SC

There are many possible combinations of N and freq to support a certain TP. For instance, $N = 16$ is used in [24] and [25]. However, there are factors, area and timing, that constrain the choice of N. We list the hardware usage in the crucial modules with N equal to 1, 2, 4, 8, 16, and 32, as shown in III. We explain how area and clock frequency constrain the N and show the optimal frequency as the following.

1) Analysis of Area: The modules listed in Table III (marked in gray in Fig. 6) dominate the area consumption when compared with other modules in SC. For these modules, area increases at a growing rate with the increase of N.

TABLE III

RELATIONSHIP BETWEEN THE CONSUMPTION OF HARDWARE RESOURCE OF CRUCIAL MODULES AND THE INCREASE OF N FROM 1 TO 32. (*: WHEN APPLYING PROPOSALS IN SECTION IV, THIS FIGURE IS 48, ELSE IT IS 96)

Module Name	FUs	Data width	Complexity	# of FU	1	2	4	8	16	32
EO/BO module	Adder	5 bits	$O(N^2)$	2.5N(N-1)	0	5	30	140	600	2480
	MUX	3 bits								
	Comparator	4 bits								
SAccs/CAccs.	Adder	Max (15bits,5bit+logN) Fig. 6(II)	O(1)	48*	48					
	Comparator	10∼15 bits								
	MUX	2 bits	O(N)	48(N-1)*	0	48	144	336	720	1488
Category Classifier	Adder	3 bits	O(N)	2N	2	4	8	16	32	64
	Comparator	8 bits	O(N)	7N	7	14	28	56	112	224

The increase in area mainly comes from the quadratic growth in quantity of function units (FUs) of EO/BO modules, listed in the second row (EO/BO module) in Table III. Since BO and EO are similar in architecture, we use EO as an example.

When $N = 1$, this sample must belong to one of the five categories. By checking the category that this sample belongs to, the corresponding accumulators for Sum and Count (SAcc and CAcc as in Fig. 6) operate. When $N = 2$, there are two cases for the second sample B that B belongs to the same category as A, or not. For the former case, the corresponding SAcc unit increments by the sum of two differences and CAcc increments by two; for the latter case, operation for each sample is the same as the case when $N = 1$. It could be noticed that the addition of difference of latter samples depends on the result of former ones, because the samples with the same category or band are accumulated together. Considering whether the rest $N - 1$ samples have the same category with the first sample or not, 2^{N-1} branches exist and $N - 1$ adders as well as multiplexers are required. Similarly, when we consider rest $N-2$ samples with the second sample, 2^{N-2} branches exist and $N-2$ adders as well as multiplexers are required. Thus, we can conclude that $N(N-1)/2$ adders, multiplexers, and comparators are totally required for each EO/BO module at N times of parallelism.

Besides, the number of inputs necessarily multiplexed to each accumulator also grows with the increase of N. As shown in Fig. 6, there are four inputs (sA to sD) for each accumulator when $N = 4$. Furthermore, the larger data width of each adder also increases the area consumption. The data are shown in the third row (SAccs/CAccs) of Table III.

2) Analysis of Timing: Though the above analysis reveals that the high area efficiency benefits from a smaller N, a smaller N means that a higher freq is required to sustain the target TP. However, the maximum frequency is constrained, since there is a loop in SAcc and CAcc, as shown in Fig. 6 (II). The path delay, mainly generated by the adder, has a lower bound irrelevant to N. Besides, the path delay in a loop cannot be reduced by pipelining. Thus, the achievable clock frequency has an upper bound, or N has a lower bound given a target TP.

3) Optimal Clock Frequency: To support 8k at 120 frames/s, the TP required equals to $7680 \times 4320 \times 120 \times 1.5$. From (8) and Fig. 8, we know that the required system frequency, 5.2 and 2.6 GHz, is higher than the maximum system frequency (1.5–1.6 GHz) when N equals to 1 and 2, respectively.

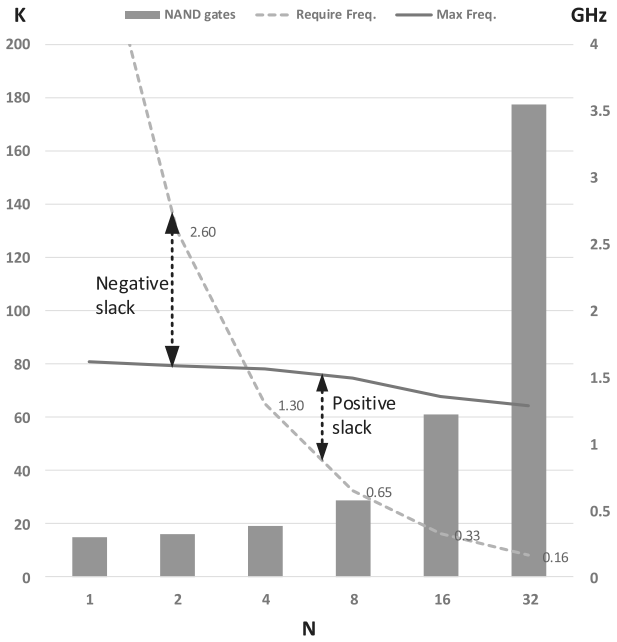


Fig. 8. Relationship among parallelism (N), the number of NAND gates of min. frequency required and max. frequency achieved in SC, under the process of SMIC 40 nm. Bars represent the approximate number of NAND gates required by the modules listed in Table III. The dashed line represents the min. frequency required, and the solid line represents the max. frequency could be achieved with various values of N.

The maximum frequency should always be larger than the frequency required to guarantee a positive slack. Thus, N equal to 4 is the optimal choice among the candidates with positive slack, since a smaller N is more area efficient. Thus, the corresponding optimal frequency is 1.3 GHz.

C. Other Feasible Models for Parallel SC

1) Multiple Sets of Accumulators: This model uses N sets of accumulators that work independently and have a final accumulation stage. It gives an $O(N)$ hardware complexity without affecting critical path, but needs $(M - 1)$ extra sets of accumulators. The hardware consumption of SAccs/CAccs in Table III increases to $O(M)$ from $O(1)$. The experiment result shows that the Accs of one category occupy 0.55k gates at 650 MHz. Thus, at least extra 13k gates are required, even with only two sets of accumulators. Our model is thus more efficient than the M-set accumulator model as long as N is less than 32.

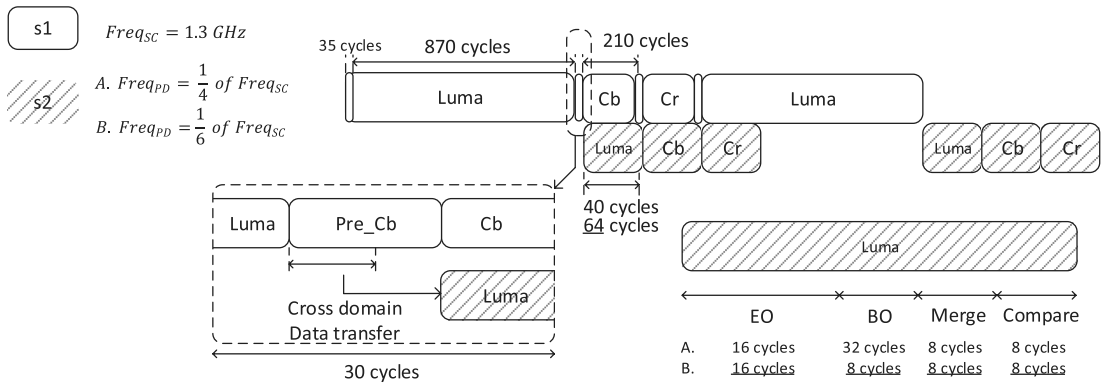


Fig. 9. Schedule of CTB processed in pipeline in two clock domains.

2) Serial: This model uses one set of registers but N sets of multiplexers and adders connected in series. It gives an $O(KN)$ hardware complexity, where $K = 48$ (four EO pattern $\times$ four categories/EO pattern $+$ 32 bands). Since each category or band needs one string of adders and MUXs connected in series. When N is small, K becomes the main influence on the area. From our analysis, this linear model is worse than our chosen model when N is not greater than 16.

Overall, compared with the above two models, the model with $N = 4$ that we choose is the most area efficient for SC.

D. Optimal Frequency in PD

The best parameter sets for each CTB are decided among four EO patterns (four categories in each), 32 bands, and two merge categories (four offsets to be evaluated in each category). Intuitively, it generates a cost for an EO category or a BO band per cycle, and a cost for each merge candidate every four cycles. This process is pipelined in three stages. Totally, it takes $16+32+4+4+4 = 60$ cycles for processing the PD of each CTB. Since the result (Sum and Count) of SC stored in registers is used by PD, the clock frequency of PD should be above a lower bound as the following equation:

$$\text{Freq}_{\text{PD}} \geq \text{NC}_{\text{PD}} * N_{\text{CTB}} * N_{\text{frame}} \tag{6}$$

where NC_{PD} is the numbers of cycles to finish PD, N_{CTB} is the numbers of CTB in a frame, and N_{frame} is the number of frames encoded in a second.

Freq_{PD} is enough to support the required TP with three pipeline stages when it is only 1/6 or 1/4 of Freq_{SC} (1.3 GHz). Their areas are 21k and 25k, respectively. When Freq_{PD} is 1/2 of Freq_{SC}, five pipeline stages are required. Its area is 30.4k gates, with the area overhead being 45%. It is very difficult to increase the frequency of PD to further match the frequency of SC, because the calculation of Offset, Dist., and Cost in PD, as shown in Fig. 7, consists of complicated multiplexing, and multiplication, which is challenging for the deep pipeline.

E. Proposed Architecture

Based on the analysis above, we propose a dual-clock architecture, where a high-speed clock drives SC and a low-speed clock drives PD, so that the features of each part could be exploited. The frequency of the high-speed clock is 1.3 GHz, and the frequency of the low-speed clock is 1/4 or 1/6 of $\text{clk}_{\text{highspeed}}$. Both of them are derived by clk_{base}. The relationship of frequency between them is shown in the following:

$$\text{clk}_{\text{high-speed}} = \text{clk}_{\text{base}} \tag{7}$$

$$\text{clk}_{\text{low-speed}} = \frac{\text{clk}_{\text{base}}}{M} = \frac{\text{clk}_{\text{base}}}{\text{floor}(NC_{\text{SC}}/\text{NC}_{\text{PD}})}. \tag{8}$$

In this paper, M is four without CRS (to be presented in Section IV) or six with CRS.

For SC, it takes 905, 240, and 240 high-speed clock cycles to process a Luma, Cb, and Cr CTB in serial, respectively. NC_{SC} is equal to 240, since the minimum number of cycles used in SC is decided by the Cb/Cr channel. For PD, it takes 60 low-speed clock cycles to find the best parameter candidates for each channel. The resulting M is 4. With CRS, M increases to 6, since the number of clock cycles for finding the best set of candidates (NC_{PD}) decreases from 60 to 36 with the number of bands decreased from 32 to 8.

The processing schedule for SC and PD is shown in Fig. 9. SC and PD are processed in pipeline stages s1 and s2, in two different clock domains, respectively. The data in s1 are kept unchanged for at least M (4 or 6) cycles, so that the data in s1 could be caught by the rising edge of low-speed clock, and be transferred to s2 before they are updated for another new CTB in s1 during this period.

It is noted that two clocks have a frequency relationship of a dividends M and are derived by clk_{base}. The rising edges of each clock are periodically aligned in delta time, making it unnecessary to have the extra data synchronization. Compared with using two completely independent clocks, it eliminates the hardware expense for additional phase-locked loops.

IV. Algorithm-Architecture Co-Optimization

A. Coarse Range Selection

The exhaustive search among 32 bands to find the best bands consumes two-third (32/48) of hardware resources of the design, while the hardware for EO dominates the rest one-third.

If we could design a hardware friendly scheme that coarsely selects the sample value ranges in each CTB to reduce the search range from 32 bands to a small number before SC, the

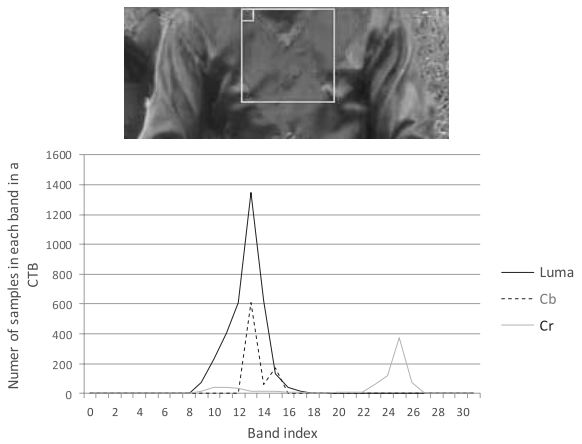


Fig. 10. Number of samples distributed in each band within a CTB (e.g., a normal CTB of video sequence of *Racehorse*_832 × 480).

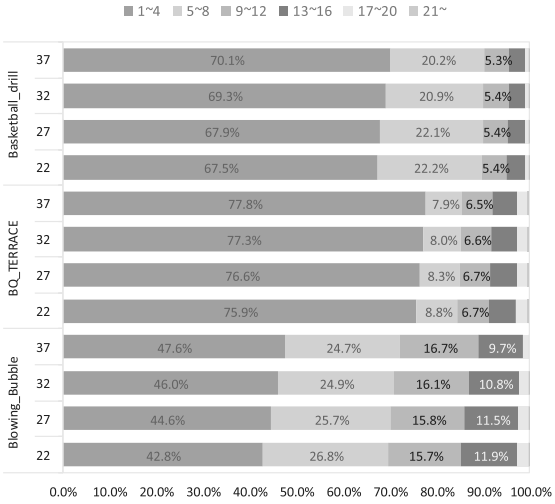


Fig. 11. Distribution of number of bands in a CTB in various sequences, *BQTerrace*, *BasketballDrill*, and *BlowingBubbles*. They are evaluated with QP of 22, 27, 32, and 37 and have the maximum, medium, and minimum BD-rate degradation, respectively, as shown in the third column of Table IV.

overall resource for collecting BO statistics can be decreased. The selected range would better cover as many samples within a CTU as possible, so that the band characteristics of a CTU could be mainly preserved. As mentioned in Section I, a BO predecision was proposed in [21]. The BO predecision scheme searches among all 32 bands to find the best one, which is software oriented and aims at speeding up the band decision process. However, it is impracticable for hardware implementation, since the limitation in SRAM bandwidth causes that lots of cycles are taken to fetch the samples in a CTU for predecision.

From Fig. 10, we observe that most of the sample values are distributed in several bands, and the distribution of the chroma samples is even more concentrated. We further collect the statistics of the number of bands used in each CTB in video sequences, so as to know how large a range of bands is enough to efficiently classify most of the samples in a CTB. The distribution of number of bands used in a CTB is shown in Fig. 11. Results show that 82% of CTBs have at least 90% of their samples concentrated in no more than eight bands. The distribution is more concentrated when the video sequence becomes larger, since each CTB tends to contain less textures. Such results indicate that the use of eight bands to collect the statistics could guarantee most of the samples to be classified.

We thus propose the CRS for BO based on most likely band estimation. It reduces both the searching space for the best bands and the relevant hardware resource for BO from 32 to 8. Before the start of SC, we define a coarse selection stage, which contains 16 cycles for a CTB. During this stage, the system makes an estimation on the bands distribution and finds the center of distribution. In each cycle of this stage, a window of 2 × 2 reconstructed samples scans 16 locations evenly distributed in a CTB. The average of the samples in the window is calculated and accumulated. In the last cycle of the preestimation stage, a final average value is calculated. The band belonging to this value is regarded as the one in the center of sample distribution within the current CTB. The left three bands and the right four bands of it are considered as the reduced eight band candidates, as shown in Fig. 12. The four consecutive bands with minimum cost from them will be selected in PD. For the pixels outside the eight ranges,

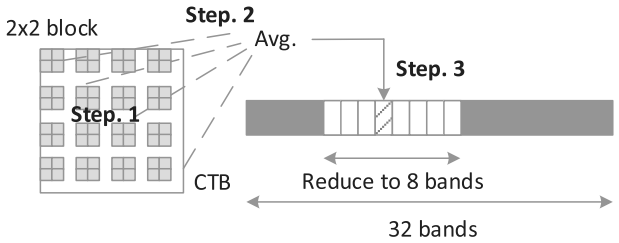


Fig. 12. Process of reduced BO candidates searching. Step 1: scan the 16 evenly distributed windows (2 × 2 reconstructed samples) one by one, calculate the average value of the samples in the window, and accumulate the average value. Step 2: estimate the average value within the current CTB. Avg. $= (\text{Sum} + 8) \gg 4$. Step 3: candidate bands range from $(\text{Avg.} \gg 3) - 3$ to $(\text{Avg.} \gg 3) + 4$.

the SC does not collect the statistics of them, which could reduce the coding efficiency. But the reduction is very limited, with average 0.2% BD rate increase in each configuration.

The experiment result shows that this proposed technique has the top-1 prediction (the proposed best band is the same as the best one from original HM) rate is about 60%, and the top-3 prediction rate is about 80%. More results about the performance of this proposed scheme evaluated in BD rate are shown in Section V.

B. Accumulator Bit Width Reduction

As shown in Fig. 6, there are 24 or 48 SAccs and CAccs (with or without CRS) in SC. Theoretically, each CAcc could increment to the maximum value, 4096 (64 × 64) for each CTB. Since a sample is classified into only one of the bands/categories, however, the classification is mutual exclusive with each other. In most cases, the final value of CAcc is about several hundreds.

We thus propose to replace the maximum value of CAcc by a smaller threshold, so as to reduce the bit width of CAcc and SAcc. Once the value in CAcc reaches a threshold, the accumulations in CAcc and the corresponding SAcc are terminated. The data width reduction of FUs in CAcc and SAcc depends on the threshold.

TABLE IV
BD RATE COMPARISON OF ALGORITHMS UNDER LDP CONFIGURATION (ANCHOR: HM-16.0, SAO OFF, AND CTB: 64 × 64)

	Video sequences in common test	SAO_On	ICIP14 [25]	Proposed without CRS or ABR	Proposed with CRS and ABR
Class A Cropped 4K×2K	Traffic	-8.1%	-8.2%	-8.2%	-8.2%
	PeopleOnStreet	-6.0%	-6.5%	-6.4%	-6.4%
	Nebuta	-7.6%	-8.8%	-8.5%	-8.4%
	StreamLocomotive	-16.0%	-17.6%	-17.6%	-17.7%
Class B 1080p	Kimono	-7.0%	-7.8%	-7.7%	-7.7%
	ParkScene	-8.1%	-8.0%	-8.0%	-8.2%
	Cactus	-11.2%	-11.7%	-11.8%	-11.6%
	BasketballDrive	-8.4%	-8.6%	-8.5%	-8.8%
	BQTerrace	-17.1%	-18.2%	-18.2%	-18.1%
Class C WVGA	BasketballDrill	-8.6%	-9.1%	-9.2%	-8.3%
	BQMall	-8.1%	-8.2%	-8.3%	-8.4%
	PartyScene	-4.9%	-4.9%	-5.0%	-5.0%
	Racehorse	-8.8%	-9.0%	-9.0%	-9.0%
Class D WQVGA	BasketballPass	-4.6%	-4.8%	-4.6%	-4.7%
	BQSquare	-4.4%	-4.4%	-4.4%	-3.7%
	BlowingBubbles	-4.2%	-4.3%	-4.3%	-4.3%
	RaceHorses	-6.1%	-6.1%	-6.0%	-6.2%
Class E 720p	FourPeople	-9.2%	-9.6%	-9.6%	-9.5%
	Johnny	-12.3%	-12.4%	-12.4%	-13.1%
	KristenAndSara	-11.2%	-12.2%	-12.1%	-12.0%
Class F	BasketballDrillText	-9.2%	-9.5%	-9.6%	-8.3%
	ChinaSpeed	-9.7%	-10.9%	-10.8%	-8.0%
	SlideEditing	-4.2%	-2.5%	-2.3%	-1.3%
	SlideShow	-7.1%	-7.8%	-8.0%	-5.5%
Class Summary	Class A	-9.5%	-10.3%	-10.2%	-10.2%
	Class B	-10.4%	-10.8%	-10.8%	-10.9%
	Class C	-7.6%	-7.8%	-7.9%	-7.7%
	Class D	-4.8%	-4.9%	-4.8%	-4.7%
	Class E	-10.9%	-11.4%	-11.4%	-11.5%
	Class F	-7.6%	-7.7%	-7.7%	-5.8%
	All	-8.4%	-8.8%	-8.8%	-8.4%

We compare the effect of BD-rate reduction among three thresholds, 1024, 2048, and 4096. The experiment result shows that there is no observable coding efficiency loss for any them. In fact, we found that a threshold of 1024 could still preserve 97% of statistics. The statistics loss has little influence on the coding efficiency.

By setting a threshold of 1024, we could reduce the data width in each CAcc as well as SAcc, contributing area reduction by 5k gates, 10% of the entire area.

V. EXPERIMENT RESULTS

The proposed design has been implemented on register transfer level in SystemVerilog. Logic synthesis and physical design have been conducted with Synopsys Design Compiler and Cadence SoC Encounter, respectively, in SMIC 40-nm CMOS standard cell library. For verification, input data and expected outputs were generated from HM 16.0 software model as stimulus and reference for the hardware design. The layout is shown in Fig. 13. The high-speed clock domain can work under the required frequency of 1.3 GHz. We evaluate the power consumption for the video sequence of *BasketballDrill* with quantization parameter (QP) =37 and under the low delay, main, P (LDP) slices only configuration. The power of our design is 48 mW when high-speed clock equals to 1.3 GHz.

To evaluate the coding efficiency of the proposed SAO, two groups of tests are conducted over the common test condition [27]. The first group of tests evaluates the following five algorithms, including the HM-16.0 default setting with SAO turned OFF (anchor) and turned ON, algorithm in [25], the proposed algorithm without and with CRS as well as ABR.

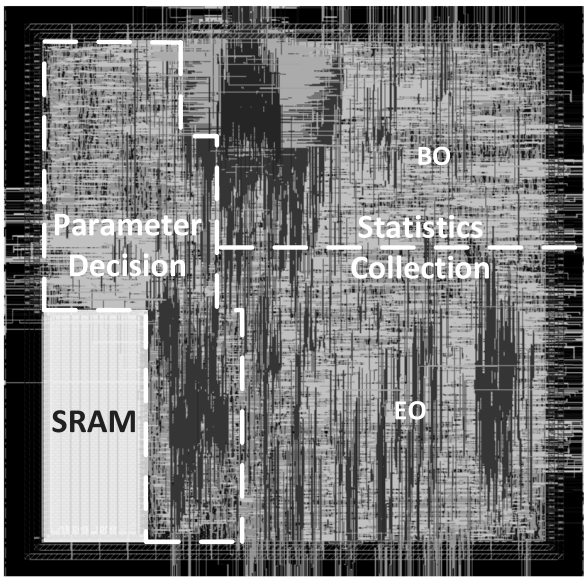


Fig. 13. Layout of the proposed SAO architecture with a core size of 0.073 mm^2 and chip-area utilization of 73%. Before the layout, the delay of the critical path is 0.66 ns, and after the layout, it is 0.76 ns. The location of PD and SC is marked, while the SRAMs for line buffer are marked in white.

TABLE V

BD RATE COMPARISON OF PROPOSED ALGORITHM WITH CRS AND ABR UNDER DIFFERENT CONFIGURATIONS, AI, RA, AND LD (ANCHOR: HM-16.0, SAO ON, AND CTB: 64 × 64)

	Video sequences in common test	AI	RA	LD
Class A Cropped 4Kx2K	Traffic	0.2%	0.2%	0.2%
	PeopleOnStreet	0.2%	-0.1%	-0.2%
	Nebuta	0.2%	-1.0%	-1.2%
	StreamLocomotive	-0.2%	-0.4%	-0.8%
Class B 1080p	Kimono	0.2%	0.1%	0.0%
	ParkScene	0.2%	0.2%	0.0%
	Cactus	0.3%	0.1%	0.1%
	BasketballDrive	0.4%	0.1%	-0.1%
	BQTerrace	-0.1%	-0.1%	-0.4%
Class C WVGA	BasketballDrill	0.7%	0.6%	0.6%
	BQMall	0.3%	0.1%	-0.0%
	PartyScene	0.1%	0.1%	-0.0%
	Racehorse	0.2%	0.0%	-0.2%
Class D WQVGA	BasketballPass	0.4%	0.1%	-0.0%
	BQSquare	0.1%	0.6%	-0.1%
	BlowingBubbles	0.1%	0.1%	-0.1%
	RaceHorses	0.2%	0.1%	-0.2%
Class E 720p	FourPeople	0.2%	0.3%	0.0%
	Johnny	0.5%	0.5%	0.2%
	KristenAndSara	0.4%	0.5%	0.0%
Class F	BasketballDrillText	0.6%	0.7%	1.8%
	ChinaSpeed	0.9%	1.3%	1.8%
	SlideEditing	0.5%	1.5%	2.7%
	SlideShow	0.6%	1.0%	1.7%
Class Summary	Class A	0.1%	-0.3%	-0.5%
	Class B	0.2%	0.1%	-0.1%
	Class C	0.3%	0.2%	0.1%
	Class D	0.2%	0.2%	0.0%
	Class E	0.4%	0.4%	0.1%
	Class F	0.7%	1.1%	2.0%
	All	0.3%	0.3%	0.3%

This group is evaluated under LDP configuration, since the effect of SAO is the most obvious [5] under this condition. The result is shown in Table IV. The second group of tests evaluates the following two algorithms, the HM-16.0 default setting with SAO turned ON (anchor) and the proposed algorithm with CRS and ABR. This group is evaluated under three configurations, all intra (AI), random access (RA), and LD. The result is shown in Table V.

Comparing our proposed design with [25] in Tables IV and VI, we know that our proposed design achieves a reduction by 69% from [25] with no coding efficiency loss in BD rate. Compared with the result the anchor in Table V under various configurations, our proposed algorithm has 0.3% coding efficiency loss in BD rate. This is achieved by the following techniques.

First, the parallelism of SC is reduced from 16 in [25] to 4 in this paper, under the requirement of meeting TP of 8k at 120 frames/s. As is shown in Section III, the cost of increasing the paralleling factor N is large in SC of SAO. The paralleling factor of 4 is much more area efficient than the factor of 16. Thus, this paper reduces the circuit area by 67%, from 156.3k to 68k, based on the same specification and algorithm setting.

Second, we found that the CRS for BO nearly has no observable loss in coding efficiency in terms of BD rate. Both the data shown in Fig. 11 and the top-3 prediction rate of 80% reveal that the range selected by our scheme covers most of the samples in a CTU. The cost of best three bands is so close that even though the best is missed by our scheme, there are second and third best bands to compensate the statistics loss. The differences on BD-rate performance among the best, second and third best bands, are small. Our proposed algorithm-architecture co-optimizations can further reduce the circuit area by 25%, from 68k to 51k, as shown in Table VII. The reduction in area is contributed by the following aspects.

1) CRS decreases the overall SAcc and CAcc in SC decrease by 50%, from 48 to 24. The registers used for storing offsets also decrease by 50%, from 48 × 3 to 24 × 3. It achieves a 13k area reduction, with an area overhead of 2k in the predecision step.
2) CRS also reduces the number of cycles for PD from 60 to 36 cycles, so that a looser time constraint for PD, increased from 2.8 to 4.8 ns, further reduces the circuit area by 3k.
3) ABR helps to reduce the data width in each CAcc and SAcc, contributing 5k gates reduction with BD rate increase of 0.1% and 0.2% in LD and LDP configuration, respectively.

When synthesized in the same Fujitsu e-Shuttle 65-nm process as in [25], our design has a logic gate count of 59.6k, with 35.5k and 24.1k for SC and PD, respectively. We give a brief comparison on power consumption by analyzing the area, frequency, and switching factor. The area–frequency product of this paper is 35.5k × 1.3G + 24.1k × 0.217G = 51.3P, 20% lower than that of [25]: 156k × 0.4G = 64.4P. Moreover, the high-parallelism SC of [25] involves the updating of more accumulator registers per clock cycle, resulting in a higher switching factor. As a result, our design is more energy efficient than [25] with reduction in both area–frequency product and switching factor.

TABLE VI

COMPARISON OF SYNTHESIS RESULT WITH THE PREVIOUS WORKS

	This work	ICIP'14 [25]	ISCAS14 [24]
Process	40nm	65nm	28nm
Area (gates)	51K	156.3K	300K
SRAM	1.14KB	1.08KB	N/A
TP (pixels/s) for encoding	4320p, 120fps	4320p, 120fps	2160p, 60fps
Cycles to finish 64×64 CTB: SC	905	384	1600
Cycles to finish 64×64 CTB: PD	40	64	N/A
Clock Freq. (MHz)	SC: 1300 PD: 217	SC & PD: 378	SC & PD: 200
Norm. TP (samples/(gates*s))	117.1K	38.2K	2.8K

TABLE VII

COMPARISON OF THE PROPOSALS WITH AND WITHOUT CRS AND ABR

		Proposed without CRS and ABR	Proposed with CRS and ABR
Area (gate)	**SC modules**	43K	30K
	PD modules	25K	21K
	Total	68K	51K
Cycles to finish 64×64 CTB :	**SC modules**	870	905
	PD modules	64	40
SRAM		1.14KB	
TP (pixels/s) for encoding		4320p, 120fps	

We have also implemented an $N = 8$, 650 MHz version to study the area and power consumption, which turns out to be 60.5k and 32 mW, respectively. The $N = 8$ version, therefore, is 20% larger in area, but 33% more efficient in power. The latter is mainly from the fact that a looser timing constraint compared with the $N = 4$, 1.3-GHz configuration now allows the synthesizer to use slower (and, therefore, smaller and less power consuming) logic cells. Moreover, we roughly implemented and estimated that the $N = 16$ version is twice in area compared with the $N = 8$ version, where an even looser timing constraint does not influence results much. Overall, the design of N larger than 8 does not show higher efficiency in energy despite being significantly larger in area. The $N = 4$ version is more efficient in area, while the $N = 8$ version is more efficient in power. Both can be taken into considerations for applications.

VI. Conclusion

SAO is a new ILF in H.265 video coding standard. Many studies have been trying to improve its performance and area efficiency of hardware design. This paper presents an efficient VLSI design of SAO estimation in H.265. We first introduce SAO and analyze its data flow. Then, we proposed the dual-clock architecture to address the heterogeneous data flows of SC and PD, by separately driving SC and PD at a high-speed clock and a low-speed clock, respectively. Two clock frequencies with a relationship of dividends M eliminate the extra hardware and implementation expense. Moreover, the algorithm-architecture co-optimizations, CRS and ABR further reduce the circuit area without observable loss in coding efficiency. Our proposed architecture occupies 51k logic gates. With a high-speed clock of 1.3 GHz and a low-speed clock of 217 MHz, 8k at 120-frames/s SAO real-time encoding can be achieved.

References

[1] T. Ito, "Future television—Super Hi-Vision and beyond," in *Proc. IEEE Asian Solid State Circuits Conf. (A-SSCC)*, Nov. 2010, pp. 1–4.

[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 Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[3] B. Bross, W. J. Han, J. R. Ohm, G. J. Sullivan, and T. Wiegand, *High Efficiency Video Coding (HEVC) Text Specification Draft 10 (for FDIS & Consent)*, document ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Joint Collaborative Team on Video Coding (JCT-VC), Jan. 2013.

[4] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[5] C.-M. Fu *et al.*, "Sample adaptive offset in the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1755–1764, Dec. 2012.

[6] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 614–619, Jul. 2003.

[7] Y.-W. Huang, T.-W. Chen, B.-Y. Hsieh, T.-C. Wang, T.-H. Chang, and L.-G. Chen, "Architecture design for deblocking filter in H.264/JVT/AVC," in *Proc. Int. Conf. Multimedia Expo (ICME)*, vol. 1. Jul. 2003, pp. I-693–I-696.

[8] K. Xu and C.-S. Choy, "A five-stage pipeline, 204 cycles/MB, single-port SRAM-based deblocking filter for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 3, pp. 363–374, Mar. 2008.

[9] Y.-C. Lin and Y.-L. Lin, "A two-result-per-cycle deblocking filter architecture for QFHD H.264/AVC decoder," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 6, pp. 838–843, Jun. 2009.

[10] D. Zhou, J. Zhou, J. Zhu, and S. Goto, "A 48 cycles/MB H.264/AVC deblocking filter architecture for ultra high definition applications," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E92-A, no. 12, pp. 3203–3210, 2009.

[11] A. Norkin *et al.*, "HEVC deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1746–1754, Dec. 2012.

[12] M. Li, J. Zhou, D. Zhou, X. Peng, and S. Goto, "A dual-mode deblocking filter design for HEVC and H.264/AVC," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E96-A, no. 6, pp. 1366–1375, 2013.

[13] W. Cheng, Y. Fan, Y. Lu, Y. Jin, and X. Zeng, "A high-throughput HEVC deblocking filter VLSI architecture for 8k × 4k application," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 605–608.

[14] M. Tikekar, C.-T. Huang, C. Juvekar, V. Sze, and A. P. Chandrakasan, "A 249-Mpixel/s HEVC video-decoder chip for 4K ultra-HD applications," *IEEE J. Solid-State Circuits*, vol. 49, no. 1, pp. 61–72, Jan. 2014.

[15] J. Zhu, D. Zhou, G. He, and S. Goto, "A combined SAO and de-blocking filter architecture for HEVC video decoder," in *Proc. 20th IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2013, pp. 1967–1971.

[16] M. Mody, N. Nandan, and T. Hideo, "High throughput VLSI architecture supporting HEVC loop filter for ultra HDTV," in *Proc. IEEE 3rd Int. Conf. Consum. Electron.-Berlin (ICCE-Berlin)*, Sep. 2013, pp. 54–57.

[17] C. M. Diniz, M. Shafique, F. V. Dalcin, S. Bampi, and J. Henkel, "A deblocking filter hardware architecture for the high efficiency video coding standard," in *Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, San Jose, CA, USA, 2015, pp. 1509–1514. [Online]. Available: http://dl.acm.org/citation.cfm?id=2757012.2757160

[18] J. Joo, Y. Choi, and K. Lee, "Fast sample adaptive offset encoding algorithm for HEVC based on intra prediction mode," in *Proc. IEEE Third Int. Conf. Consum. Electron.-Berlin (ICCE-Berlin)*, Sep. 2013, pp. 50–53.

[19] J. Joo and Y. Choi, "Dominant edge direction based fast parameter estimation algorithm for sample adaptive offset in HEVC," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 3749–3752.

[20] Y. Choi and J. Joo, "Exploration of practical HEVC/H.265 sample adaptive offset encoding policies," *IEEE Signal Process. Lett.*, vol. 22, no. 4, pp. 465–468, Apr. 2015.

[21] G. Chen, Z. Pei, Z. Liu, and T. Ikenaga, "Low complexity SAO in HEVC base on class combination, pre-decision and merge separation," in *Proc. 19th Int. Conf. Digit. Signal Process. (DSP)*, Aug. 2014, pp. 259–262.

[22] F. Rediess, R. Conceição, B. Zatt, M. Porto, and L. Agostini, "Sample adaptive offset filter hardware design for HEVC encoder," in *Proc. IEEE Vis. Commun. Image Process. Conf.*, Dec. 2014, pp. 299–302.

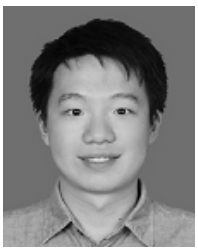
[23] F. Rediess, R. Conceição, B. Zatt, M. Porto, and L. Agostini, "Cost function optimization and its hardware design for the sample adaptive offset of HEVC standard," in *Proc. 22nd Eur. Signal Process. Conf. (EUSIPCO)*, 2014, pp. 206–210.

[24] M. Mody, H. Garud, S. Nagori, and D. K. Mandal, "High throughput VLSI architecture for HEVC SAO encoding for ultra HDTV," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Jun. 2014, pp. 2620–2623.

[25] J. Zhu, D. Zhou, S. Kimura, and S. Goto, "Fast SAO estimation algorithm and its VLSI architecture," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 1278–1282.

[26] J. Zhu, D. Zhou, S. Kimura, and S. Goto, "Fast SAO estimation algorithm and its implementation for 8k × 4k 120 FPS HEVC encoding," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E97-A, no. 12, pp. 2488–2497, 2014.

[27] F. Bossen, *Common Test Conditions and Software Reference Configurations*, document JCTVC-F900, JCT-VC, Torino, Italy, Jul. 2011.



Jianbin Zhou received the B.E. degree in software engineering from the South China University of Technology, Guangzhou, China, in 2012, and the M.E. degree in engineering from Waseda University, Kitakyushu, Japan, in 2014, where he is currently pursuing the Ph.D. degree in engineering.

His current research interests include VLSI architectures and algorithm for multimedia signal processing.

Dr. Zhou was a recipient of the Ting Hsin Scholarship in 2013 and 2014.

Dajiang Zhou (S'08–M'10) received the B.E. and M.E. degrees from Shanghai Jiao Tong University, Shanghai, China, and the Ph.D. degree in engineering from Waseda University, Kitakyushu, Japan, in 2010.

He is currently an Assistant Professor with the Graduate School of Information, Production and Systems, Waseda University. His current research interests include algorithms and implementations for multimedia and communications signal processing, especially in low-power high-performance VLSI architectures for video codecs, including H.265/HEVC and H.264/AVC.

Dr. Zhou received a number of awards, including the Best Student Paper Award of the VLSI Circuits Symposium 2010, the International Low Power Design Contest Award of ACM ISLPED 2010, the 2013 Kenjiro Takayanagi Young Researcher Award, and the Chinese Government Award for Excellent Students Abroad of 2010. He was a recipient of the Research Fellowship of the Japan Society for the Promotion of Science from 2009 to 2011. His work on an 8K UHDTV video decoder VLSI chip received the 2012 Semiconductor of the Year Award of Japan.

Shihao Wang received the B.E. degree in electronics engineering from Shanghai Jiao Tong University, Shanghai, China, in 2012, and the M.E. degree in engineering from Waseda University, Kitakyushu, Japan, in 2014, where he is currently pursuing the Ph.D. degree in engineering.

His current research interests include VLSI architectures and algorithm for multimedia signal processing.

Dr. Wang was a recipient of Ting Hsin Scholarship in 2013 and 2014.

Shuping Zhang received the B.E. degree from the Beijing Institute of Technology, Beijing, China, in 2011, and the M.E. degree from Waseda University, Kitakyushu, Japan, in 2014, where he is currently pursuing the Ph.D. degree.

His current research interests include 3DLSI implementations for video coding technologies.

Takeshi Yoshimura (M'86) received the B.E., M.E., and D.Eng. degrees from Osaka University, Osaka, Japan, in 1972, 1974, and 1997, respectively.

He joined NEC Corporation, Kawasaki, Japan, in 1974, where he has been engaged in research and development efforts devoted to computer application systems for communication network design, hydraulic network design, and VLSI computer-aided design (CAD). From 1979 to 1980, he was on leave with the Electronics Research Laboratory, University of California at Berkeley, Berkeley, CA, USA, where he was involved in VLSI CAD layout. He is a Professor with Waseda University, Kitakyushu, Japan,

Prof. Yoshimura received the Best Paper Awards from the Institute of Electronics, Information and Communication Engineers of Japan and the IEEE CAS Society. He is a member of the Information Processing Society of Japan.

Satoshi Goto (LF'11) received the B.E. and M.E. degrees in electronics and communication engineering and the D.Eng. degree from Waseda University, Kitakyushu, Japan, in 1968, 1970, and 1978, respectively.

He joined Central Research Laboratories of NEC, Kawasaki, Japan, in 1970, where he was involved in LSI design, multimedia system, and software as the General Manager and Vice President. Since 2002, he has been a Professor with the Graduate School of Information, Production and Systems, Waseda University. He is a Visiting Professor with Shanghai Jiao Tang University, Shanghai, China, and Tsinghua University, Beijing, China.

Prof. Goto is a fellow of the Institute of Electronics, Information and Communication Engineers. He is a member of the Science Council of Japan. He served as a General Chair of the ICCAD, ASPDAC, VLSI-SOC, ASICON, and ISOCC and was a Board Member of the IEEE CAS Society.