IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 24, 2019, accepted October 8, 2019, date of publication October 11, 2019, date of current version October 24, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2946907*

# A Compact 32-Pixel TU-Oriented and SRAM-Free Intra Prediction VLSI Architecture for HEVC Decoder

**YIBO FAN, GENWEI TANG, AND XIAOYANG ZENG, (Member, IEEE)**
State Key Laboratory of ASIC and System, Fudan University, Shanghai 201203, China

Corresponding author: Yibo Fan (fanyibo@fudan.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61674041, in part by the Alibaba Innovative Research (AIR) Program, in part by the IBM Faculty Award, in part by the Innovation Program of Shanghai Municipal Education Commission, in part by the Pioneering Project of Academy for Engineering and Technology, and in part by the Fudan-CIOMP Joint Fund.

**ABSTRACT** In the High Efficiency Video Coding (HEVC), a variety of CU sizes and intra prediction modes significantly improve coding efficiency, but also bring higher computational complexity. This paper proposes a new compact VLSI architecture for HEVC intra prediction, which is geared towards 8K video decoding. It supports all the transform unit (TU) sizes and 35 HEVC intra prediction modes. First, this paper introduces a TU-oriented intra predictor with a throughput of 32 pixels, which can be newly arranged with the TU size. It can be a line of 32 pixels, two lines of 16 pixels, 4 lines of 8 pixels or four lines of four pixels. This TU-oriented architecture allows intra-prediction and inverse discrete cosine transform (IDCT) to be computed in parallel, removing the memory between them. In addition, a horizontal and vertical line buffer for reference sample is proposed, which only cost 0.8K bit and is implemented by register files with SRAM-free. Finally, to further reduce hardware consumption, multipliers can be shared in the prediction. The implementation result shows that the compact architecture supports 8K video application and costs 66.2K logic gates, which is synthesized with the TSMC 65nm process under 400MHz.

**INDEX TERMS** 32-pixel, TU-oriented, SRAM-free, HEVC intra prediction, 8K decoder.

## I. INTRODUCTION

The Joint Collaborative Team on Video Coding (JCT-VC) has proposed the latest video coding standard, High Efficiency Video Coding (HEVC) [1]. It can achieve about a 50% bit-rate reduction than the previous video coding standard, H.264/AVC [2], but with higher computational complexity.

Like AVC, HEVC also uses block-based coding. Each video frame is divided into multiple CTUs, and each CTU is further divided into smaller CUs. And in HEVC, the CU is further divided into PUs and TUs, where PU is the basic prediction unit and TU is for transform and quantization.

The maximum CU size in HEVC is $64 \times 64$, not $16 \times 16$ in AVC, but the maximum size of TU is only $32 \times 32$. The PU size of intra prediction is suitable for TU, so the largest PU is $32 \times 32$. In addition, the number of HEVC intra prediction modes is 35, not 9 in AVC, in order to describe the texture better [4]. Although the complex block partition scheme and the prediction mode can improve the coding quality, the VLSI implementation is more challenging for high-throughput and high-resolution video coding.

Due to the higher complexity and data dependencies, it is difficult to propose a compact high-performance intra prediction architecture. Previously, there was some related work on the intra prediction VLSI architecture, and some were optimized for the reference pixel preparation and prediction process. Zhou *et al.* [5] introduced a LUT-based reference sample fetching scheme to reduce the reference sample number and the chip area. Their final VLSI architecture can fulfill 8K UHDTV video decoding. In addition, for better reference samples fetching efficiency, a hierarchical memory and mode-dependent intra smoothing architecture is proposed by Huang *et al.* [6], which could also support $3840 \times 2160$ videos decoding. A unified reference sample indexing scheme is also illustrated by Jiang *et al.* [7] to avoid the reference re-arrangement. There are lots of works focusing on optimizing the prediction process. A full pipelined intra prediction VLSI architecture is introduced by Min *et al.* [8],

The associate editor coordinating the review of this manuscript and approving it for publication was Liang-Bi Chen.

where the reference buffer will be traversed by depth-first and the scanning order will be, which could generate 4 pixels/cycles for 3840 × 2160 videos application. For higher throughput, an 8K HEVC intra prediction architecture is proposed by Zhou *et al.* [9], with pipelined 4 × 4 block and cyclic SRAM, whose throughput is 15 samples/cycle. Zhou *et al.* [10] showed a new reference pixel-mapping method to reduce hardware cost. There are some works focusing on other interesting points. Kalali *et al.* [11] introduced a low energy architecture with computation reduction techniques. To reduce the computational complexity of 4 × 4 PU, Abramowski *et al.* [12] created a separate path to process the 4 × 4 PU prediction with little increase of hardware consummation, which is also adopted in [13]. Liu *et al.* [14] proposed a reconfigurable intra prediction architecture to avoid the transpose register array which can support 2560 × 1600 video coding. Liu *et al.* [15] presented a highly pipelined architecture using a post-order traversal method to reduce the internal buffer.

In previous related work, the intra prediction was a fixed-shape structure. However, the output shape of the IDCT varies with the size of the TU, so the inverse discrete cosine transform (IDCT) and intra prediction are divided into two pipeline stages. In order to solve the problem of output data mismatch, this paper proposes several techniques to solve these problems.

1) A high-throughput intra prediction architecture is proposed to meet the needs of high-speed 8K video codec, which is twice the previous work.

2) The shape of the 32-pixel can be changed with the TU, and the output format of the IDCT is the same so that the intra prediction and the IDCT can be processed in parallel, and the ping-pong memory between the two can be removed, thereby greatly reducing the use of the on-chip memory.

3) After analyzing the intra prediction algorithm, a 0.8K bit intra prediction reference sample buffer is proposed, which greatly reduces the storage space of the reference sample. In addition, a fast and compact method based on zig-zag position is proposed to calculate whether the reference sample is valid. Under this situation, it is unnecessary to apply the SRAM to buffer the reference samples, so the circuit area and cycle for retching reference samples can be significantly reduced.

4) In order to reduce hardware cost, a multiplier sharing scheme is proposed for Planar and angular mode, which can still be competitive in hardware cost with higher throughput.

The next steps in this paper are as follows. Section II introduces the motivation for proposing the architecture, followed by a brief introduction to HEVC intra prediction, including reference pixel management and prediction formulas in section III. Section IV will be the proposed TU-oriented hardware architecture and the horizontal and vertical line buffer. Section V will be the result of the implementation and comparison with other work, and Section IV will summarize this article.

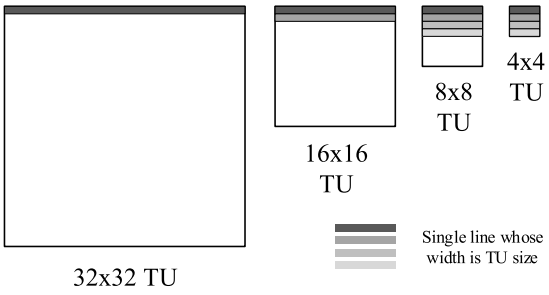



**FIGURE 1.** IDCT output format of different TUs.

## II. MOTIVATION

Due to the higher algorithm complexity and diverse CU sizes, the high throughput of intra prediction and IDCT is important for high-performance HEVC decoders. For intra prediction, 16 sample/cycle may be required to meet the demand [5], while inverse discrete cosine transform (IDCT) may require 32 samples/cycle to meet the demand [17]. The high throughput requirements create many difficulties for VLSI architecture design. At the same time, higher processing speeds require more hardware consumption and implementation complexity.

### A. LIMITATION OF IDCT OUTPUT FORMAT

In HEVC, the TU size varies from 4×4 to 32×32, and the output shape of the IDCT also varies with the TU size [17]–[20]. For the 32-sample IDCT in [17], the output shape may be a row of 32 samples of 32 × 32 TU, two rows of 16 samples of 16 × 16 TU, 4 rows of 8 samples of 8 × 8 TU, or 4 rows of 4 samples of 4 × 4 TU. Although this design can lead to higher throughput, there are many difficulties in reading and writing data. In response to this, Fan *et al.* [21] proposed a parallel-access storage processing method, which provides an effective data access method for DCT and solves the storage difficulties caused by multiple TU sizes.

However, since the shape of the intra prediction is mostly fixed 4 × 4, which does not match the output of the IDCT, the ping-pong memory between the two is inevitable. Therefore, we want to optimize the intra prediction structure to match the IDCT output format and eliminate unnecessary memory.

### B. LIMITATION OF HIGH-THROUGHPUT HEVC DECODER

The IDCT and the prediction are two main modules of the decoder, where the residues plus the prediction values are the reconstructed values for decoded video. In the previous works relating to the HEVC decoder system, the intra prediction and IDCT are placed into two pipeline stages for higher throughput, such as [16], [22], [23]. In the previous works, the IDCT was designed to be a 32-sample TU-oriented architecture where the IDCT output format will vary with the TU size. However, the intra prediction was designed to be a fixed 4 × 4 format, so the throughput and format between the IDCT and

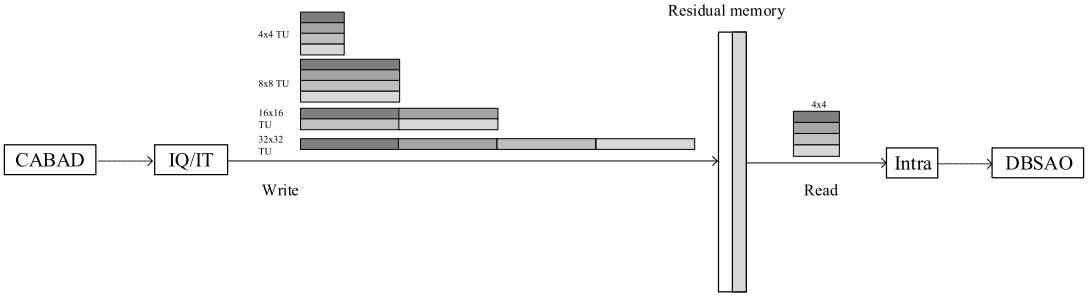



**FIGURE 2.** The brief pipeline diagram of the HEVC intra decoder in the previous works.

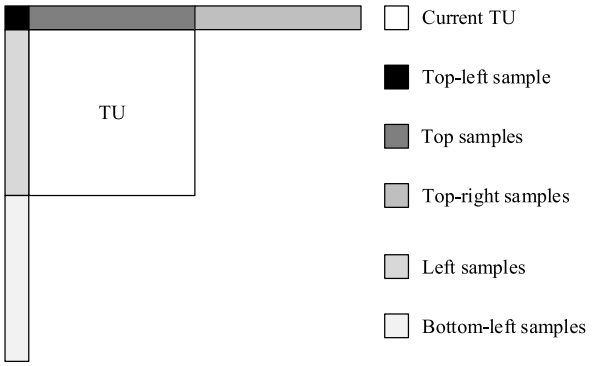



**FIGURE 3.** Reference samples.

intra prediction did not match. The ping-ping memory for the residues have to be applied to buffer the residues from IDCT.

Although the divided intra prediction and IDCT can work in higher throughput, it is inevitable to apply additional ping-pong memory to buffer the residual data after IDCT, and the memory size can be more than 100K bit for a largest coding unit (LCU). The memory will be a huge cost for the HEVC decoder system. The brief HEVC intra decoder system of the previous works is depicted as the Fig.2. The CABAD means the context-adaptive binary arithmetic coding decoder, and IQ/IT means the inverse quantization and transform, and the DBSAO means the Deblocking and the sample adaptive offset. They are the different coding tools of the HEVC standard.

If the output format of intra prediction can be in agreement with the IDCT as this work proposed, they can work in parallel. In such a case, the decoder architecture can not only fulfill the requirements of high throughput but also avoid the additional ping-pong residual memory to reduce the hardware cost.

### C. REFERENCE SAMPLE MEMORY

In the HEVC intra prediction, the computation relies on the reference samples which are extracted from the neighbor decoded TUs, as shown in Fig. 3. To simplify the reference samples fetching logic, we often buffer them with SRAM. In the previous works, the reference sample memory may be very large with all the reference buffered. However, it may be unnecessary. In the current TU/PU prediction, only the neighbor reference will be useful. And some reference samples will be used only once, such as the top samples of the current TU in Fig. 3. Based on this, we can reduce the size of the reference samples memory to save the chip area.

## III. HEVC INTRA PREDICTION

In HEVC, the intra encoding is to remove the spatial redundancy. The intra prediction is calculated by the reconstructed pixels around the current CU/TU. There are two key steps to be performed, reference samples preparation and the sample prediction for every TU.

### A. REFERENCE SAMPLES PREPARATION

The reference samples are used to predict current TU and it is extracted from the neighbor CUs/TUs, which are already decoded previously in the same frame or slice. The reference samples are positioned at the top-left, top, top-right, left, bottom-left of the current TU where there are [4 * (TU_size) + 1] pixels in total, as shown in Fig. 3.

Due to the fixed processing order, the TUs around the current TU may not be fully decoded. In this case, some reference may not be available. The first is that the reference pixels at all locations are not available, in which case the reference pixels are filled with $2^{\mathrm{bit}Depth-1}$. The second is that if only partial reference pixels are not available, then these reference pixels will be replaced by reference pixels at those valid positions. If all the reference pixels are available at all locations, all reference pixels are extracted from the surrounding TUs directly.

After the reference samples are built, it will also be filtered, when the TU size is 8 × 8, 16 × 16, and 32 × 32 under some prediction modes. There are two kinds of filtering in the HEVC standard, [1 2 1]/4 and bilinear filter. If the prediction mode is DC, the filter will be disabled, and if Planar, the filter is necessary. For the angular mode, using filter or not is decided by the specific prediction mode. For example, the 2, 18 and 34 will be filtered by normal filter in 8 × 8 TU. And for the 16 × 16 TU, the filter will be performed to most of the mode, except for the 9, 10, 11, 25, 26 and 27. The filter for 32 × 32 TU is more critical, where only 10 and 26 will be avoided, and the strong filter may be applied under some specific cases.

### B. INTRA PREDICTION

The calculation of intra prediction is based on the prepared reference pixels and the current prediction mode. There are a total of 35 intra prediction modes in HEVC, which can be divided into three categories, DC, Planar and angular mode. Different prediction modes have different calculation formulas. In all the formulas below, the TU size is specified by the *nTbS*, which is 4, 8, 16, 32, and reference sample array is specified by the $p[x][y]$, which is prepared.

In the DC mode, the prediction value is the average of the top and left reference samples, as (1). In addition, the result may be weighted with the reference pixel to smooth the block edge, when the prediction position is adjacent to the reference sample.

$$dcVal = \Big(\sum_{x=0}^{nTbS-1} p[x][-1] + \sum_{y=0}^{nTbS-1} p[-1][y] + nTbS\Big) >> ((\log_2 nTbS) + 1) \quad (1)$$

In the Planar mode, the prediction values are calculated by the four reference samples located at the top, left, top-right and bottom-left as the equation (2).

$$\begin{aligned} &predSamples[x][y] \\ &= ((nTbS-1-x)^{*}p[-1][y]+(x+1)^{*}p[nTbS][-1] \\ &\quad + (nTbS-1-y)^{*}(p[x][-1]) \\ &\quad + (y+1)^{*}p[-v1][nTbS]+nTbS) >> ((\log_2 nTbS)+1) \end{aligned} \quad (2)$$

For angular mode, the prediction samples are calculated based on the equation (3) whose mode value is less than 18, where the *ref[x][y]* is extracted from the reference samples according to the angular mode and the *predAngle* is also related to the prediction mode.

$$\begin{aligned} predSamples[x][y] = ((32 - iFact)^{*}ref[y + iIdx + 1] \\ + iFact^{*}ref[y + iIdx + 2] + 16) >> 5 \end{aligned} \quad (3)$$

where

$$\begin{aligned} iIdx &= ((x+1)^{*}predAngle) >> 5 \\ iFact &= ((x+1)^{*}predAngle) \,\&\, 31 \end{aligned}$$

To be careful, if the prediction mode is equal or greater than 18, the $y$ of *ref[]* will be replaced by $x$ and the $x$ of *iIdx* and *iFact* will be substituted by $y$. The angular mode prediction formula differs with the prediction mode.

## IV. PROPOSED VLSI ARCHITECTURE

As an important part of the HEVC decoder, the intra prediction architecture should be compatible with the whole system. Due to the limited working frequency of high-throughput decoder, the number of the cycle for processing a single CTU is limited. In addition, the IDCT often be implemented by folded architecture to save hardware cost and energy, such as [16], [19]. Therefore, to be consistent with IDCT, the intra prediction for every TU will be performed in serial, and the luma and chroma samples will also be processed in serial to reduce the cost.

Moreover, the 32-pixel intra prediction puts a lot of pressure on the preparation of reference pixels and pixel prediction. The prediction will require more hardware resources, and variable predictive shapes can make hardware design more complex.

In order to solve these problems, this paper proposes several key technologies. The first is a 0.8K bit reusable horizontal and vertical buffer to store the reference pixels, which is SRAM-free. This can greatly reduce the access time of the reference pixels. In addition, in order to reduce the hardware cost of the prediction formula, two multiplier sharing modules are proposed.

### A. TU-ORIENTED ARCHITECTURE

In order to match the processing of IDCT, this paper adopts the intra prediction structure of parallel processing, which can make the process control simpler. Figure 4 depicts the proposed intra prediction architecture. Within a CTU, TUs will be processed in the order of zig-zag. Based on the position of the TU within the current CTU, the position of the reference sample is checked for validity and the corresponding reference sample preparation will be performed.

In order to improve the preparation efficiency of the reference pixels, we will prepare reference pixels for all possible prediction modes, not mode-adaptive. After the reference pixel preparation is completed, 32-pixel intra prediction will be performed, and the corresponding prediction formula is selected according to different modes, as shown by the process element (PE) in Fig.4. During the prediction process, 32 PEs will be rearranged according to the size of the TU. The predicted pixel and the residual after the IDCT are then added to obtain the reconstructed pixel. The reconstructed pixels located in the last column or the last row of the current TU will be updated into the reference sample buffer and used as reference samples by the next TU prediction.

The Fig.5 is the proposed space-time diagram. For all TUs, *VA* and *OUT* only require one cycle because our reference pixel storage uses register files instead of SRAM. But for *RP*, the required cycle will vary with the size of the TU. For instance, as the statement in section III, the $4 \times 4$ TU is filter-free, so there is no need to perform the filter process for $4 \times 4$ TU and the cycle for the $4 \times 4$ TU filter can be saved. Therefore, the reference sample preparation of $4 \times 4$ TU will take only one cycle, and for the other TU size, it will take two cycles to finish reference preparation. The additional cycle is to filter the samples if necessary. As for the *PRE*, the required cycle will change with the TU. Because proposed 32-pixel intra prediction, we only need one cycle for $4 \times 4$ TU, two cycles for $8 \times 8$ TU, eight cycles for $16 \times 16$ TU and thirty-two cycles for $32 \times 32$ TU. The 32-pixel prediction will be calculated in parallel and there are 32 prediction logics to compute the values for every sample. The $X$ and $Y$ of prediction in the Fig.4 illustrate the position of the current TU

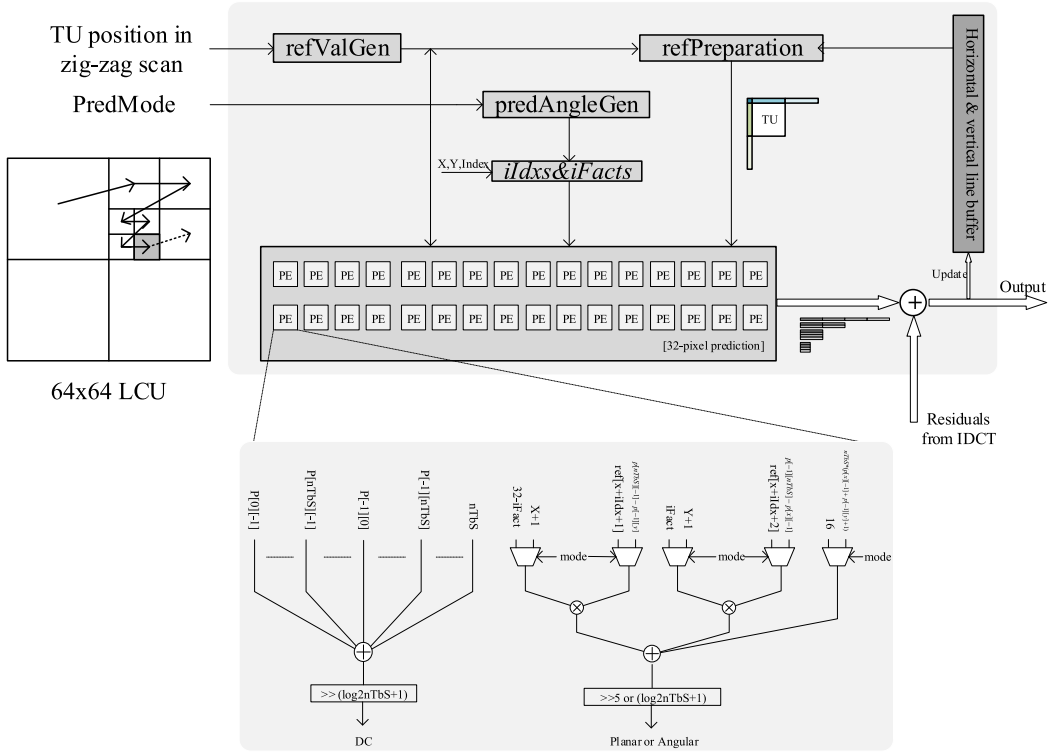



**FIGURE 4.** The proposed intra prediction architecture.

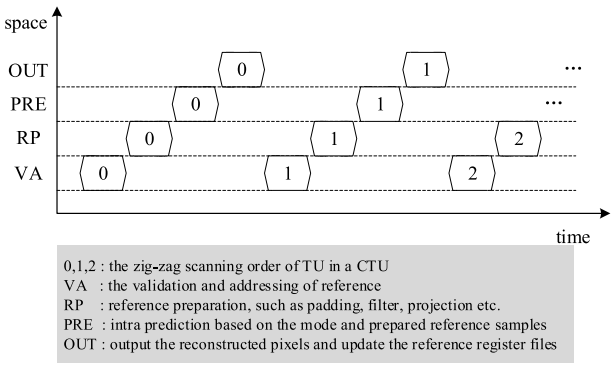



**FIGURE 5.** Diagram of TU processing order.

in the LCU, and the *Index* presents the line number of current 32-pixel prediction. For example, *Index* = 0 of 32 × 32 TU means the 32-pixels prediction is the first line of 32 × 32 TU and the *Index* of the second line will be changed to 2.

In a CTU, the worst case is that all TU sizes are 4 × 4 because its average required cycle for 4 × 4 block is more than the other TU size. For example, the 4 × 4 TU requires four cycles to complete the prediction, and for 8 × 8 TU, it will take six cycles as the previous statement. For Y:U:V=4:2:0 there are 256 4 × 4 blocks of Y, 64 4 × 4 blocks for U and 64 4 × 4 blocks for V, so the required cycle will be 4 * (256+64+64) = 1536 to complete the whole CTU. In this worst case, the video decoding requirements of 8K@30fps can still be met. In other cases, the intra prediction architecture proposed in this paper can also meet the decoding requirements of 8K video.

Based on the proposed architecture, we can achieve high-throughput video coding. The processing speed is faster than most of the previous works whose architectures only support 4K application.

### B. REUSABLE HORIZONTAL & VERTICAL LINE BUFFER

As the statement in Section III, the reference samples are located at the five positions around the current TU. Therefore, the sample will be saved when it will be referenced by other TUs. The reference samples that are not used will be ignored, which will save the buffer size occupied by the reference pixels.

Based on the analysis, we propose the horizontal and vertical line buffer to register all the useful reference samples, where there are 64 horizontal samples, 32 vertical samples, and 8 top-left samples. Because the luma and chroma are processed serially, they will use the same buffer. Although there may be some duplication between them, for example, the top left corner may overlap with horizontal or vertical pixels. However, this can greatly reduce the access complexity of the reference pixels and is easy to implement. There is an example of the horizontal and vertical line buffer in Fig.6 (a).

For the current TU in Fig.6, the top-right and bottom-left reference samples in TU3 and TU5 have not been decoded yet, so they cannot be used. Therefore, the reference samples for current TU will be extracted from the top, left, top-left

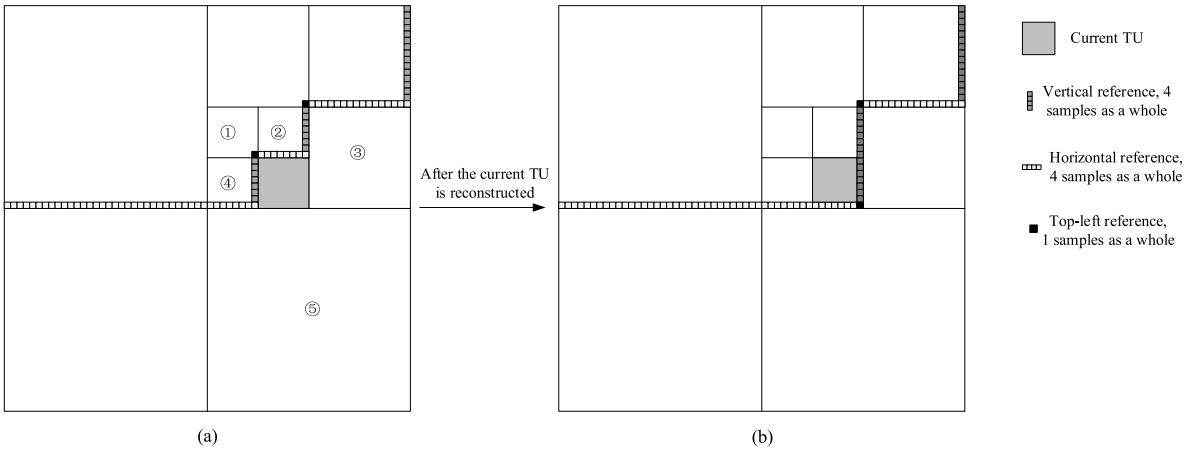



**FIGURE 6.** Example of updating the horizontal and vertical line buffer.

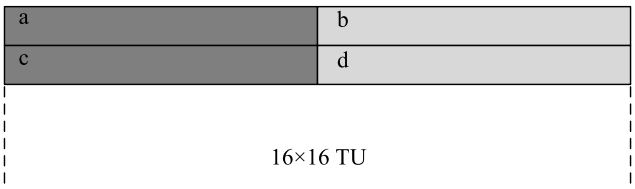



**FIGURE 7.** iIdx and iFact computation of 16 × 16 TU.

TU. And the valid reference samples are already buffered in the proposed horizontal and vertical line buffer. Moreover, the reference sample buffer is also applicable to all the rest TU processing in the LCU.

After the current TU processing is completed, the state of the horizontal and vertical line buffer will be changed to Fig.6 (b) for the next TU prediction. The new reference samples will be registered and the useless samples will be covered by the useful reference samples. The useless reference samples will not be referred to by the remaining TU prediction, so they can be removed and replaced by the new reference samples to save the memory size for the reference samples. By the memory size calculation, only 0.8K bit, we can notice that buffer size is smaller than the previous works, and it is unnecessary to implement it with SRAM which is timing and resource-consuming. The register files are better suited to the higher speed and simpler fetching operation, where the reference samples can be read or written in only one cycle to save time for reference fetching.

In addition, whether the reference samples are valid or not can also be quickly obtained by zig-zag scanning order. For example, in the Fig.6, the 4 × 4 block is used as the basic calculation unit, and the current TU position is 108 in the order of zig-zag. In this case, the zig-zag order of the other five positions is also easily available. If the zig-zag order of the reference sample position is smaller than the position of the current TU, the reference sample of the corresponding position is valid, and vice versa. The reference position validation can be generated quickly and easily based on the zig-zag scanning order.

### C. MULTIPLIER SHARING

As we all know, the intra prediction relies on the sample position. The different position will be computed by different parameters. Hence, the 32-pixel prediction will be performed by 32 computation logic in parallel. And for the different TU size, only the parameters in the formula will be changed. This computation is the same as the previous works.

However, the higher throughput will increase resource consummation. Some ideas to save the multiplier focused on the TU-oriented prediction are proposed.

First, equation (2) of Section III can be rearranged to (4). Since the limitation of *nTbS*, 4, 8, 16, 32, we will use the shift operation instead of the third. In this case, only two multipliers will be required, the same number of multipliers as the angular mode calculation. Therefore, we can combine two prediction formulas and reuse their multiplier.

$$\begin{aligned} &predSamples[x][y] \\ &= ((x+1)*(p[nTbS][-1]-p[-1][y]) \\ &\quad +(y+1)*(p[-1][nTbS]-p[x][-1]) \\ &\quad +nTbS*(p[x][-1]+p[-1][y]+1)) \\ &>> ((\log_2 nTbS)+1) \end{aligned} \tag{4}$$

Second, the *iIdx* and *iFact* calculation can also share their multiplier, where the result of (x+1)*predAngle in (3) can be reused. In addition, we also don't need to use multipliers to calculate all the results of 32 *iIdxs* & *iFacts*. We just take the 16 × 16 TU whose prediction mode is less than 18 as an example. The eight *iIdxs* & *iFacts* in block *a* can be computed by multipliers, and the 8 *iIdxs* & *iFacts* of *b* can be inferred by adder, and block *c* and *d* is equal to *a* and *b* respectively. For the other TU size, their parameter computation can also be inferred based on this idea. For example, the 32 × 32 TU only need to compute the left 8 positions with the multiplier, and the remaining 24 samples can be generated by the adder. As to the prediction mode is greater than 18, it is identical to the idea, and the only difference is to swap the *x* and *y*.

**TABLE 1. Comparison with previous works.**

| | This work | Min[8] | Huang[6] | Zhou[5] |
|---|---|---|---|---|
| Mode | All | All | All | All |
| Size | All | All | All | All |
| Process | 65nm | FPGA | 40nm | 90nm |
| Area | 66.2K | 14.0K LUT 5.5K Reg | 27.0K | 72.1K |
| Ref. | 0.8Kb | 6.0Kb | 4.9Kb | 21.0Kb |
| Throughput (sample/cycle) | 32 | 4 | 2 | 15 |
| Frequency (MHz) | 400 | 110 | 200 | 397 |
| Core Power (mW) | 5.43 | N/A | 2.11 | N/A |
| Norm. TP. (sample/cycle/k-gate) | 0.48 | N/A | 0.07 | 0.21 |
| Norm. Power (mW/sample) | 0.17 | N/A | 1.06 | N/A |
| Video decoding (fps) | 8K@30 | 4K@30 | 4K@30 | 8K@120 |

## V. IMPLEMENTATION RESULT

The final design is implemented with Verilog and we synthesize it by TSMC65nm process under 400MHz working frequency. Finally, the implementation result shows that it can support 8K video decoding at 30 fps.

Table.1 is the implementation results and the comparison with others. All of them in the list can process all the prediction modes and TU sizes of HEVC intra prediction, which means they are more practical. From the table, it can be figured out that the throughput of our proposed 32-pixel intra prediction is more than twice their works, such as [5] which can also support the 8K application. The parallel architecture in [5] can improve the frame rate, but it does not match the folded IDCT architecture. What's more, the buffer size for reference samples is far less than previous work, only 15% of the latest work [8]. Although the 32-pixel prediction will increase the logic cost, the proposed architecture can save the resources. Therefore, the final logic area normalized power consumption is also competitive compared with them. For the research described in [6], the throughput is less than our work, because our work can process 8K video decoding. The normalized power consumption or the normalized throughput can show better performance.

The comparison above just takes the intra prediction into consideration, but the impact on the HEVC decoder system should also be considered. Although the TU-oriented architecture has made it more complex and the frame rate is lower than [5] with the same resolution, it can greatly benefit the HEVC decoder system. The reason for the lower frame rate is that our proposed architecture is processed serially to be consistent with the IDCT architecture, where the [5] is in parallel. By the proposed intra prediction architecture, the IDCT and prediction can be performed in parallel to save the memory for residuals which may be more than 100K bits.

## VI. CONCLUSION

This work has proposed a compact 32-pixel TU-oriented HEVC intra prediction VLSI architecture, targeting at 8K video decoding. The TU-oriented prediction output format can make the IDCT and intra prediction work in parallel to save area of the HEVC decoder. Because of the 0.8K bit horizontal and vertical line buffer for the reference samples and the two multiplier sharing scheme, we can significantly reduce the cost. Therefore, we can support the 8K video decoding with less logic and smaller memory size for both intra prediction and the HEVC decoder system.

In future work, the proposed techniques can also be applied to the next video coding standard VVC/H.266. For example, the proposed variable prediction shape is suitable for multiple CU size in the VVC, which can be rectangle. The line-based prediction in VVC will simplify the computation of *deltaInt & deltaFract*, which is only related to the position $y$. The proposed horizontal and vertical line buffer for reference sample can also be applied to the VVC. Although the VVC adopts more intra prediction modes and coding sizes, the reference samples are also extracted from the five positions, top, top-left, top-right, left, and bottom left. Therefore, the line buffer can also be used in VVC, where each pixel should be registered separately for the variable reference length in VVC.

## REFERENCES


[1] 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.

[2] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, 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.

[3] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, "Block partitioning structure in the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1697–1706, Dec. 2012.

[4] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, "Intra coding of the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1792–1801, Dec. 2012.

[5] J. Zhou, D. Zhou, H. Sun, and S. Goto, "VLSI architecture of HEVC intra prediction for 8K UHDTV applications," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Dec. 2014, pp. 1273–1277.

[6] C. T. Huang, M. Tikekar, and A. P. Chandrakasan, "Memory-hierarchical and mode-adaptive HEVC intra prediction architecture for quad full HD video decoding," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 7, pp. 1515–1525, Jul. 2014.

[7] Y. Jiang, D. Llamocca, M. Pattichis, and G. Esakki, "A unified and pipelined hardware architecture for implementing intra prediction in HEVC," in *Proc. Southwest Symp. Image Anal. Interpretation*, Apr. 2014, pp. 29–32.

[8] B. Min, Z. Xu, and R. C. C. Cheung, "A fully pipelined hardware architecture for intra prediction of HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 2, pp. 2702–2713, Dec. 2017.

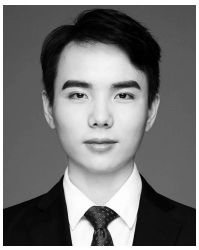
[9] J. Zhou, D. Zhou, S. Wang, T. Yoshimura, and S. Goto, "High performance VLSI architecture of H. 265/HEVC intra prediction for 8K UHDTV video decoder," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. 98, no. 12, pp. 2519–2527, 2015.

[10] W. Zhou, Y. Niu, X. Lian, X. Zhou, and J. Yang, "A stepped-RAM reading and multiplierless VLSI architecture for intra prediction in HEVC," in *Proc. Pacific Rim Conf. Multimedia*. Cham, Switzerland: Springer, 2016, pp. 469–478.

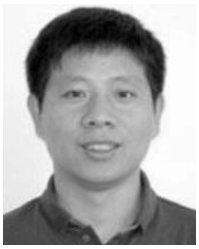
[11] E. Kalali, Y. Adibelli, and I. Hamzaoglu, "A high performance and low energy intra prediction hardware for HEVC video decoding," in *Proc. Conf. Design Archit. Signal Image Process.*, Oct. 2012, pp. 1–8.

[12] A. Abramowski and G. Pastuszak, "A double-path intra prediction architecture for the hardware H.265/HEVC encoder," in *Proc. 17th Int. Symp. Design Diagnostics Electron. Circuits Syst.*, Apr. 2014, pp. 27–32.
[13] G. Pastuszak and A. Abramowski, "Algorithm and architecture design of the H.265/HEVC intra encoder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 210–222, Jan. 2016.
[14] Z. Liu, D. Wang, H. Zhu, and X. Huang, "41.7 BN-pixels/s reconfigurable intra prediction architecture for HEVC 2560 × 1600 encoder," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 2634–2638.
[15] C. Liu, W. Shen, T. Ma, Y. Fan, and X. Zeng, "A highly pipelined VLSI architecture for all modes and block sizes intra prediction in HEVC encoder," in *Proc. IEEE 10th Int. Conf. ASIC*, Oct. 2013, pp. 1–4.
[16] D. Zhou, S. Wang, H. Sun, J. Zhou, J. Zhu, Y. Zhao, J. Zhou, S. Zhang, S. Kimura, T. Yoshimura, and S. Goto, "An 8K H.265/HEVC video decoder chip with a new system pipeline design," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 113–126, Jan. 2017.
[17] P. K. Meher, S. Y. Park, B. K. Mohanty, K. S. Lim, and C. Yeo, "Efficient integer DCT architectures for HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 1, pp. 168–178, Jan. 2014.
[18] H. Sun, D. Zhou, J. Zhu, S. Kimura, and S. Goto, "An area-efficient 4/8/16/32-point inverse DCT architecture for UHDTV HEVC decoder," in *Proc. IEEE Vis. Commun. Image Process. Conf.*, Dec. 2014, pp. 197–200.
[19] J. Goebel, G. Paim, L. Agostini, B. Zatt, and M. Porto, "An HEVC multi-size DCT hardware with constant throughput and supporting heterogeneous CUs," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 2202–2205.
[20] M. Tikekar, C.-T. Huang, V. Sze, and A. Chandrakasan, "Energy and area-efficient hardware implementation of HEVC inverse transform and dequantization," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 2100–2104.
[21] Y. Fan, L. Huang, Y. Bai, and X. Zeng, "A parallel-access mapping method for the data exchange buffers around DCT/IDCT in HEVC encoders based on single-port SRAMs," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 12, pp. 1139–1143, Dec. 2015.
[22] 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.
[23] M. Abeydeera, M. Karunaratne, G. Karunaratne, K. De Silva, and A. Pasqual, "4K real-time HEVC decoder on an FPGA," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 236–249, Jan. 2016.

**YIBO FAN** received the B.E. degree in electronics and engineering from Zhejiang University, Hangzhou, China, in 2003, the M.S. degree in microelectronics from Fudan University, Shanghai, China, in 2006, and the Ph.D. degree in engineering from Waseda University, Tokyo, Japan, in 2009. He was an Assistant Professor with Shanghai Jiao Tong University, Shanghai, from 2009 to 2010. He is currently an Associate Professor with the College of Microelectronics, Fudan University. His research interests include image processing, video coding, and associated VLSI architecture.



**GENWEI TANG** received the B.S. degree in microelectronics from Wuhan University, Wuhan, Hubei, China, in 2017. He is currently pursuing the M.S. degree in microelectronics with Fudan University, Shanghai, China. His research interests include VLSI design, algorithms, and VLSI architectures for multimedia signal processing.



**XIAOYANG ZENG** (M'05) received the B.S. degree from Xiangtan University, Xiangtan, China, in 1992, and the Ph.D. degree from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, China, in 2001. From 2001 to 2003, he was a Postdoctoral Researcher with Fudan University, Shanghai, China. He joined the State Key Laboratory of ASIC and System, Fudan University, as an Associate Professor, and he is currently a Full Professor and the Director. His research interests include information security chip design, system-on-chip platforms, and VLSI implementation of digital signal processing and communication systems.