doi:10.3772/j.issn.1006-6748.2019.01.010

Parallelization of intra prediction algorithm based on array processor①

Zhu Yun (朱 筠)*, Jiang Lin②**, Shi Pengfei***, Xie Xiaoyan***, Shen Xubang****
(*School of Microelectronics, Xidian University, Xi'an 710071, P. R. China)
(**Xi'an University of Science and Technology, Xi'an 710054, P. R. China)
(***School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, Xi'an 710121, P. R. China)
(****Xi'an Microelectronic Technology Research Institute, Xi'an 710065, P. R. China)

Abstract

For the characteristics of intra prediction algorithms, the data dependence and parallelism between intra prediction models are first analyzed. This paper proposes a parallelization method based on dynamic reconfigurable array processors provided by the project team, and uses data level parallel (DLP) algorithms in multi-core units. The experimental results show that Y-component of peak signal to noise ratio (Y-PSNR) is improved about 10dB and the time is saved 63% compared with high-efficiency video coding (HEVC) test model HM10.0. This method can effectively reduce codec time of the video and reduce computational complexity.

Key words: high-efficiency video coding (HEVC), intra prediction, parallelization mapping

0 Introduction

With the development of multimedia technology and the demand for video quality, the Video Joint Coding Task Force (JCT-VC) began to develop high-efficiency video coding (HEVC) standards in 2010 [1]. Compared with H.264, HEVC saves 50% bit rate at the same image quality. However, the computational complexity of HEVC has increased more than three times, and requires a large amount of data transmission. Therefore, how to optimize HEVC effectively is the focus of research.

Many experts and scholars have proposed various optimization solutions. With the combination of the transform domain edge detection method and mode search steps at the neighborhood, Ting and Chang[2] could reduce the probable modes significantly. Kim et al.[3] analyzed the correlation between coding unit (CU) and transform unit (TU), and simplified the CU quadtree partitioning, which could effectively reduce the coding complexity. In Ref.[4], the efficiency of intra-coding was proposed to reduce the number of prediction patterns according to the distribution characteristics of CU pixels. Jiang et al.[5] proposed a method which also took the advantage of Bayesian decision to obtain thresholds, by which early CU pruning could be determined. Some fast algorithm[6] was at the expense of a certain degree of calculation accuracy and video quality of the case, reducing the coding time, and improving the efficiency of computing. The fast algorithm made accurate prediction by heuristically reducing the number of candidates for rate distortion optimization (RDO) process[7]. According to different scene requirements to choose a different fast algorithm, you should carefully analyze the characteristics of the video image before selecting the algorithm, e.g. block segmentation optimization algorithm for low-resolution video images and the fast selection algorithm for high-resolution video images, so that various algorithms would play their own advantages as much as possible. With the development of multi-core computing architecture, parallel coding is an effective solution to achieve high computational complexity of encoders[8-10]. Due to the natural parallelism of the hardware platform, it has the advantages of low design cost and low power consumption. A novel hardware solution that reduces the complexity of DMMs for 3DHEVC was presented in Ref.[11]. The parallel design of the intra prediction on the parallel platform is an effective means to improve the efficiency of video coding and meet the real-time coding[12,13]. This paper uses the array processor provided by the project team as a platform to study the parallel model from the forecasting model.

① Supported by the National Natural Science Foundation of China (No. 61772417, 61634004, 61602377, 61272120), the Shaanxi Provincial Co-ordination Innovation Project of Science and Technology (No. 2016KTZDGY02-04-02) and the Shaanxi Provincial key R&D plan (No. 2017GY-060).
② To whom correspondence should be addressed. E-mail: jianglin@xust.edu.cn
Received on Apr. 9, 2018

1 Related work

At present, the implementation of the encoder platform is the following two: general-purpose processor and hardware accelerator. However, the general processor has insufficient support for computationally intensive programs, and hardware accelerators are less flexible in handling. In this paper, the dynamic reconfigurable array processor is parallel to implementation of intra prediction algorithm.

1.1 Array processor simulation platform

The dynamic reconfigurable array processor is developed by the project team. It supports H.264/AVC, MVC, H.265/HEVC and other video codec, including different profiles and levels. The processor is composed of the 1 024 PEs in the form of adjacent interconnection, only 8 × 8 PEs is shown in Fig. 1, including a hierarchical configuration network (HCN), a global controller, and other parts. The size of each PE instruction and data sharing storage can be adjusted dynamically. The array processor divides 4 × 4 PEs into a processing cluster (PEG) logically. The PE has a dataflow mode which is enabled or disabled by the control flow coming from HCN and deriving from global controller.

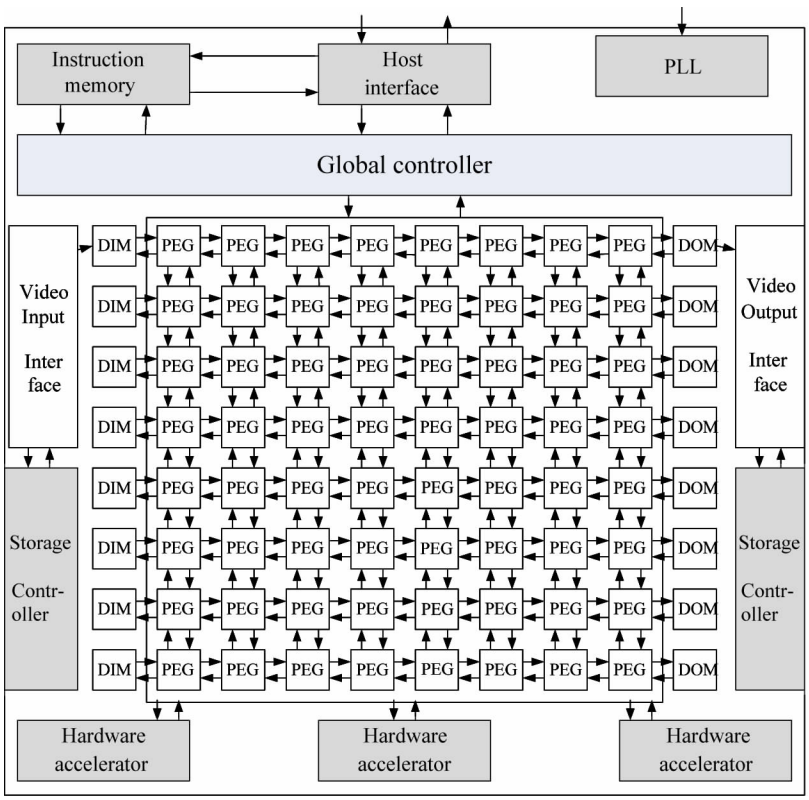

Fig. 1 Top structure of array processor

Each PE contains 16 registers, including 12 local registers and 4 shared registers. One of the data interaction methods adopted by PEG is the adjacency interconnection structure. PE can access each other by shared register and PE in four directions. The four directions include shared registers RE, RW, RS and RN respectively, as shown in Fig. 2. There are two ways of data interaction between PEs. Mode 1 is shown in the dashed arrow, and mode 2 is shown in the solid arrow. Mode 1 is used to send data from the local registers R3, R4, R5, and R6 directly to the execution units of the neighboring processing units as the source operations. Mode 2 that is the transfer of data from local PE to adjacent registers R3, R4, R5 and R6 through shared register RE, RS, RW and RN, and the data from the corresponding register is operated in the subsequent processing.

Another kind of data interaction is the distributed shared storage structure under the unified addressing mode, and the data exchange in PEG can also be realized through the high speed switching unit. The following intra prediction parallel implementation has used these two kinds of data interactive modes.

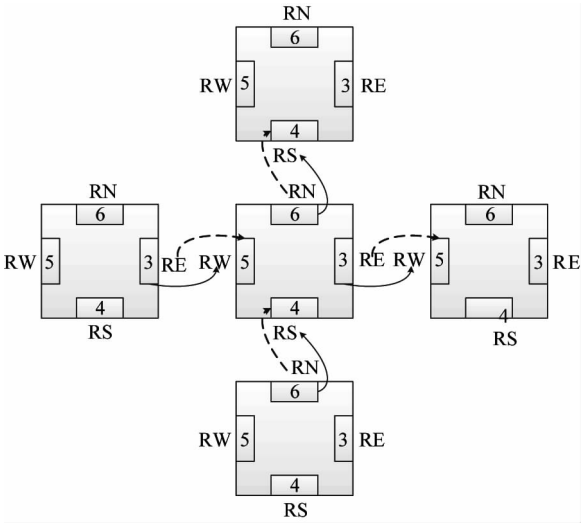

Fig. 2 Adjacent interconnection in PEG

1.2 Intra prediction parallelism analysis

HEVC provides a flexible quadtree division for intra prediction blocks, including coding tree units (CTU), coding units (CU), and predict units (PU). Each frame image is first divided into non-overlapping parallel CTUs, which can be recursively partitioned into smaller CUs by means of quadtree segmentation structures. The PU is the basic unit for carrying the intra prediction information. In Fig. 3(a), the arrowed dashed line indicates the order in which coding units are traverses. Fig. 3(b) shows the corresponding quadtree structure, where "1" indicates downward decomposition, and "0" indicates no decomposition. The "1" and "0" sequences of "10001010" represent the quadtree structure of the coding tree unit. It is worth noting that when the size of the coding unit is minimum 8×8, it is not necessary to use "1" or "0" to indicate whether to continue the decomposition. Because the data processing in the video algorithm is basically performed with $N \times N$ rectangular blocks, this array processor architecture is more suitable and more effective.

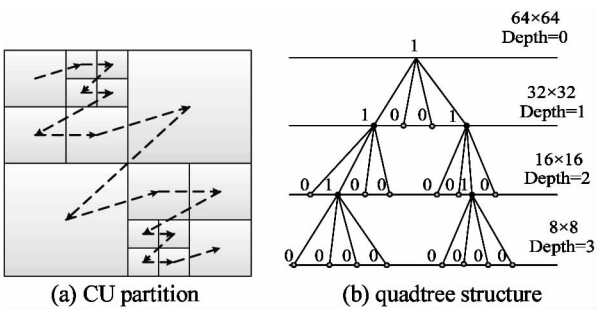

Fig. 3 Coding tree unit

HEVC provides up to 35 predictive patterns which are 33 angle mode, DC mode and Planar mode, as shown in Fig. 4. By increasing the number of prediction directions, the prediction direction of intra prediction is finer and more accurate, and the spatial redundancy can be removed more effectively. The angle is small where the angle approaches the horizontal left, or vertical upward direction, the angle is large where it is near the diagonal direction. The numbers in Fig. 4 indicate the mode number corresponding to each prediction direction. Letter H is used to represent the horizontal axis and the following numbers represent that the projections are offset from the horizontal. V is used to indicate the vertical axis; the following number indicates the deviation of the predicted direction from the vertical upward direction.

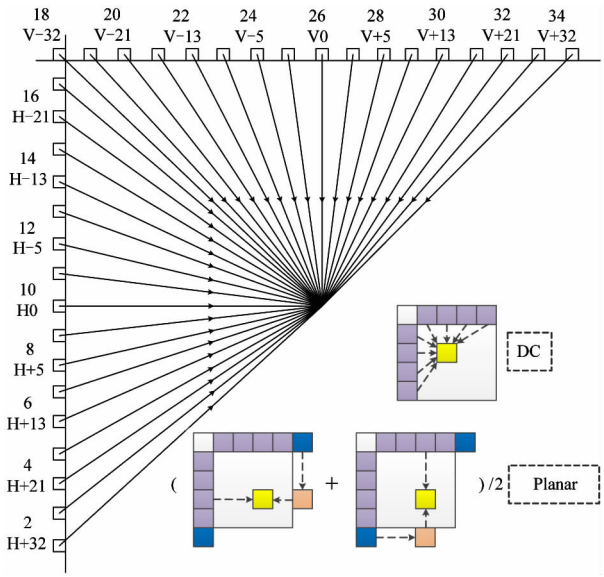

Fig. 4 Intra prediction patterns

The fine-angle prediction mode makes prediction more accurate, thereby reducing transform-coded prediction residuals. These processes have improved the quality of video coding, but the computational complexity and resource consumption cannot be ignored. The prediction mode in intra prediction has a certain probability, and not all prediction modes are the optimal prediction modes.

This paper analyzes and compares the results of CU segmentation and mode selection for different test sequences. The statistical results are shown in Fig. 5 and Fig. 6. It can be seen that in the process of mode selection, DC, Planar and some angle modes occupy a large proportion. Therefore, DC, Planar, Ang-26 (Vertical), Ang-22, Ang-20, Ang-18, Ang-10 (Horizontal), and Ang-2 were selected as the prediction mode to determine the optimal mode. This paper chooses 8×8 block as an example to describe the design idea.

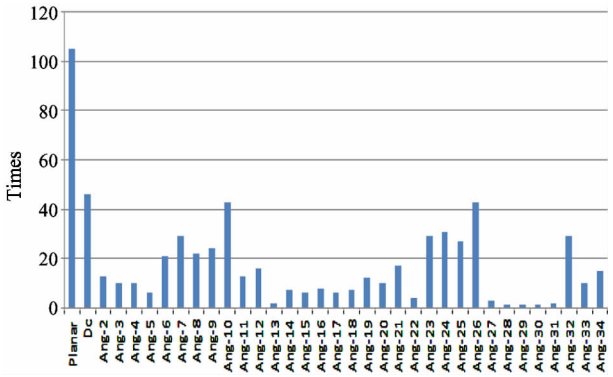

Fig. 5 176 × 144 intra prediction mode chart

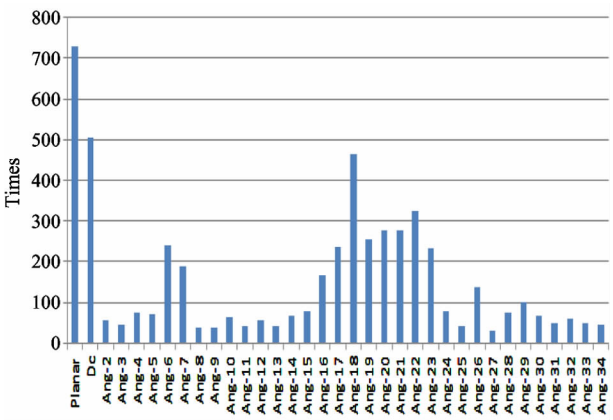

Fig. 6 832 × 480 intra prediction mode chart

2 Intra prediction parallelization scheme

Fig. 7(a) shows a parallelization scheme for processing 8 × 8 prediction blocks using the PE array. DIM is a data buffer for caching video sources. The hollow arrows indicate that the data communicates in the adjacent interconnection. Virtual arrows and real arrows indicate that the data is accessed using shared storage.

Firstly, each frame image is divided into 8 × 8 blocks from the cache, and then 8 × 8 blocks are allocated to the corresponding processing unit, and each processing unit performs a mode process. When all predictions are completed, the optimal prediction model is calculated and passed to the next level. The data flow diagram is shown in Fig. 7(b).

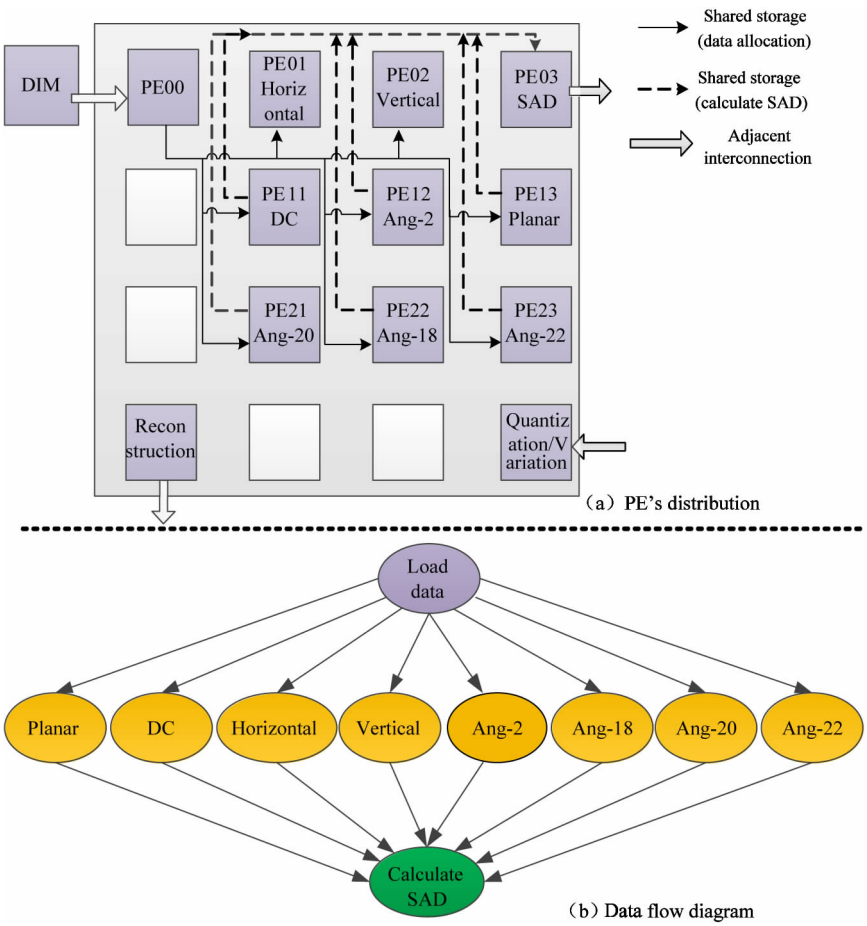

Fig. 7 Intra prediction algorithm mapping and data flow

The process of mapping the intra prediction algorithm to the PE array is as follows.

Step 1 Data loading

PE00 accesses DIM through adjacent interconnection of register R10, reading the corresponding original pixel and reference pixel. Each encoded block needs to read 97 pixel values, as shown in Fig. 8. The 64 pixels in the gray rectangle are original values and the 33 pixels in the black box are reference values. The order of data loading is first the current block proceeds from left to right, top to bottom, followed by the reference pixels. It could first load the reference pixel (named $P[-1][-1]$) above the current block, as shown in Fig. 9. The problem of edge blocks needs to be considered because there are no reference pixel blocks around the edge blocks. According to the interpolation method in the HEVC standard, the reference pixel of the edge blocks is set to 128. For the middle blocks, left upper, top and left sides of the current blocks are their reference blocks.

1	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128
2	128	36	36	26	26	28	30	32	34	23	25	28	30	32	30	31	25	26
3	128	108	100	100	106	106	100	110	110	96	96	101	99	97	100	100	94	93
4	128	133	129	133	125	121	127	121	125	123	123	121	121	121	125	123	129	115
5	128	127	123	121	126	126	123	127	119	125	119	117	112	110	111	115	125	118
6	128	119	122	120	123	125	124	128	118	125	118	117	118	119	121	121	125	122
7	128	118	119	125	122	124	131	117	131	123	122	124	121	120	117	119	127	109
8	128	123	122	124	126	126	124	128	124	121	119	119	119	119	117	114	111	107
9	128	128	123	123	119	121	129	123	127	115	118	121	121	122	121	123	125	109
10	128	127	114	115	117	119	120	117	115	124	119	123	115	111	108	109	109	108
11	128	119	120	123	120	118	120	113	115	124	113	123	117	117	121	119	109	108
12	128	115	122	119	120	120	119	117	125	118	123	115	116	114	112	113	107	120
13	128	121	128	124	120	120	121	125	113	120	123	119	121	121	118	119	109	108
14	128	113	123	116	123	121	114	114	127	109	116	121	121	120	119	120	121	105
15	128	127	117	122	118	120	122	114	127	115	121	122	119	116	115	116	119	109
16	128	119	123	122	120	120	122	118	125	115	117	118	117	116	115	116	115	109
17	128	117	129	124	117	117	128	128	127	113	115	117	117	116	116	117	121	111
18	128	115	119	122	123	123	123	121	113	116	113	114	113	112	113	115	114	107

Fig. 8 Storage sketch of video sequence in DIM

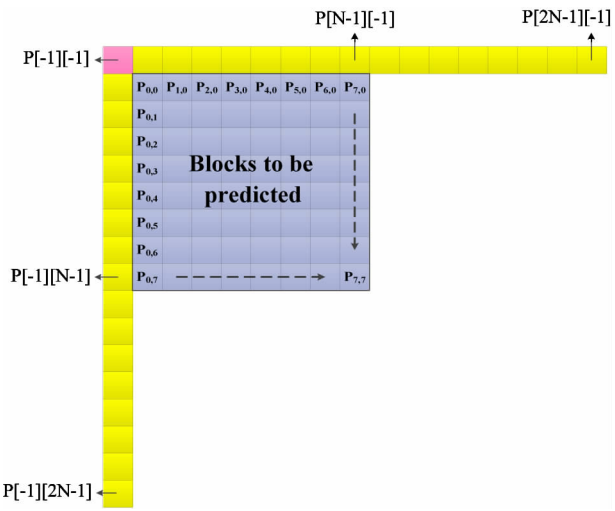

Fig. 9 Reference pixel coordinate display

Step 2 Data Distribution

PE00 needs to assign these values to different PEs after reading pixel values. In order to reduce time consumption and improve efficiency, the choice of distribution mode is shared storage.

Step 3 Parallel prediction

Fig. 7(a) shows that each PE performs a different prediction processing mode. The current block pixels is stored in 0 – 63 address of each PE data memory, whose storage order is from top to bottom and from left to right. The top and upper left pixels of the current block are stored in 64 – 80 address. The left pixels of current block are stored in 81 – 96 address. Some predictive models may produce discontinuous pixel-level faults at block boundary after prediction, especially for horizontal and vertical models in DC and angle prediction. The parallel processing of these prediction models is described below.

Planar mode: The horizontal component prediction is carried out firstly. The data is loaded into PE13 from eight left side of a column reference pixels and the upper right reference (named $P[N][-1]$), and then implemented $(N-x)\times P[0][y]+1)\times P[N][-1]$. Secondly, the vertical component prediction is performed $(N-y)\times P[0][y]+(y+1)\times P[-1][N]$ from the eight top row reference pixels and the lower left reference (named $P[-1][N]$). Finally, the horizontal component, vertical component and 8 are plused, and then right shift 4 bits, the overall prediction is made.

DC mode: First, PE11 reads the left and top reference pixels of current block from the data memory and calculates an average denoted *dcValue*. Next, pixel $P[0][0]$ applies a three-tap $[1\ \ 2\ \ 1]/4$ smoothing filter to do $P[-1][0]+2\times dcValue+P[0][-1]+2)>>2$ operation. Then, the first line prediction is completed by execution $(P[x][0]+3\times$ dcValue $+2)$ and $1\leqslant x\leqslant N-1$. The first column pre-

diction is completed by execution ($P[0][y] + 3 \times dcValue + 2) >> 2$ and $1 \leqslant y \leqslant N-1$. Finally, the remaining pixels are all predicted using *dcValue*.

Horizontal mode: PE01 reads eight left column reference pixels from the data memory. Proceed to the right in the horizontal direction until the prediction of 8 pixels per line is completed. The post-processing of the horizontal mode is required according to Eq. (1).

$$P[x][0] = P[x][0] + (P[x][-1] - P[-1][-1] >> 1) \quad (1)$$

Vertical mode: PE02 reads the top 8 reference pixels without the upper left pixel from the data memory. Predictions are performed in a vertical sequence until 8 pixels per column are completed. The post-processing of the vertical mode is carried out according to Eq. (2).

$$P[0][y] = P[0][y] + (P[-1][y] - P[-1][-1] >> 1) \quad (2)$$

Step 4 SAD calculation

PE03 calculate the SAD value according to Eq. (3), where S_A is the original pixel value, S_B is the predicted value, i or j is equal respectively from 0 to $N-1$.

$$SAD(i, j) = \sum_{i,j} | S_A(i, j) - S_B(i, j) | \quad (3)$$

Step 5 Mode selection

Each predictive model passes the respective SAD values to PE03 through shared storage, and then the optimal prediction mode will be selected by judging SAD. According to the optimal prediction model, the residual value is passed to the next stage (transform / quantization algorithm) by adjacent interconnection.

So far, the implementation of intra prediction algorithm in video coding has been completed. The full video encoder also needs other members of the project team to coordinate the implementation of other algorithms.

3 Experimental results

To make it easier to verify the video array processor, the working group developed a simulation platform. It is built on the Questasim platform and uses the system-level modeling language System C and HDL language simulation software, which includes assembler, tools, debugging modules and other parts. It could display the PE clock period, PE operation of the register information, memory information provided by Questasim. It is used to verify the correctness of the video parallel algorithm design scheme, and the basic functions of the software.

To evaluate the proposed intra prediction parallelism, five test sequences with different resolutions but the same QP are compared, as shown in Table 1. Compared to the HM10.0 with 'All Intra' configuration, the Y-PSNR is improved about 10dB and the time is saved about 63%. Compared with the Ref. [14], the time is saved about 50%.

Table 1 Comparison of coding time and Y-PSNR

Sequences	HM10.0		Ref. [14]	This paper	
	Y-PSNR(dB)	Time(ms)	Time(ms)	Y-PSNR(dB)	Time(ms)
Traffic 2560×1600	63.68	28948.25	15684.16	74.21	7237.06
Kimono1 1920×1080	65.57	15927	8629.25	74.58	3981.75
Basket ball Drill Text 832×480	65.71	2989	1619.44	75.73	747.25
Race Horses 416×240	63.61	685.75	371.54	74.12	171.44
Carphone_qcif 176×144	35.41	85.13	46.12	45.45	16.76
Average	58.80	6541.63	5270.10	68.82	2430.85

The project group uses the dynamic reconfigurable array processor to perform the function simulation of the intra-frame coding of the test sequence carphone_qcif.yuv. The video screenshot is shown in Fig. 10. From the image quality point of view, the encoding results of array processor (as shown in Fig. 10(b)) are much better than the HM10.0 encoding results (as shown in Fig. 10(a)).

The project team implements HEVC intra-frame coding based on array processors, and uses assembly modules to improve and optimize key computation-intensive key algorithm modules, which not only im-

proves overall coding efficiency, but also improves video image quality. Since each key module (including intra prediction, deblocking filtering, quantization transformation, inverse quantization inverse transformation, etc.) is optimized, the encoded pixel value is very close to the original video pixel value, and the deviation is small, so the overall video quality is improved.

(a) HM10.0 encoding (b) Array processor encoding

Fig. 10 Coded image comparison

4 Conclusion

Through the analysis and statistics of the intra prediction, 8 kinds prediction models with higher probability are selected. In this paper, using a dynamic reconfigurable array processor, each PE handles one prediction mode, and the prediction calculations among the 8 modes are independent of each other. The prediction of each pixel does not affect each other. The problem of predict mode calculation and pixel point calculation waiting in serial mode is solved, so that multiple modes can be processed in parallel when processing one mode. This parallel scheme greatly saves the time required for data loading and mode prediction when a single processing unit is serially implemented, and improves computational efficiency and resource utilization.

Reference

[1] Clare G, Henry F, Pateux S. JCTVC-F274: Wavefront parallel processing for HEVC encoding and decoding [EB/OL]. http://phenix.int-evry.fr/jct/doc-end-user/documents/6_Torion/wg11/JCTVC-F174-V2.Zip: Phenix, 2014

[2] Ting Y C, Chang T S. Fast intra prediction algorithm with transform domain edge detection for HEVC[C]. In: Proceedings of the 2012 IEEE Asia Pacific Conference on Circuits and Systems, Kaohsiung, China, 2012. 144-147

[3] Kim J, Yang J, Lee H, et al. Fast intra mode decision of HEVC based on hierarchical structure[C]. In: Proceedings of the 8th International Conference on Information, Communications & Signal Processing, Singapore, 2011. 1-4

[4] Wang L L, Siu W C. Novel adaptive algorithm for intra prediction with compromised modes skipping and signaling processes in HEVC[J]. *IEEE Transactions on Circuits & Systems for Video Technology*, 2013, 23(10): 1686-1694

[5] Jiang G Y, Yang X X, Peng Z J, et al. Fast CU depth range selection and early CU pruning for HEVC[J]. *Optics & Precision Engineering*, 2014, 22(5):1322-1330

[6] Zhao L, Zhang L, Ma S, et al. Fast mode decision algorithm for intra prediction in HEVC[C]. In: Proceedings of the 2011 Visual Communications and Image Processing, Tainan, China, 2011. 300-304

[7] Yan S, Hong L, He W, et al. Group-based fast mode decision algorithm for intra prediction in HEVC[C]. In: Proceedings of the 8th International Conference on Signal Image Technology and Internet Based Systems, Sorrento, Italy, 2013. 225-229

[8] Choi K, Jang E S. Leveraging parallel computing in modern video coding standards[J]. *IEEE Multimedia*, 2012, 19(3):7-11

[9] Yan C, Zhang Y, Dai F, et al. Efficient parallel framework for HEVC deblocking filter on many-core platform [C]. In: Data Compression Conference, IEEE Computer Society, Snowbird, USA, 2013. 530

[10] Yan C, Zhang Y, Dai F, et al. Highly parallel framework for HEVC motion estimation on many-core platform[C]. In: Data Compression Conference, Snowbird, USA, 2013. 63-72

[11] Amish F, Bourennane E B. An efficient hardware solution for 3D-HEVC intra-prediction[J]. *Journal of Real-Time Image Processing*, 2017. 1-13. DOI: 10.1007/s11554-016-0664-1

[12] Zhou M, Sze V, Budagavi M. Parallel tools in HEVC for high-throughput processing[C]. In: Applications of Digital Image Processing XXXV, San Diego, USA, 2012, 849910-849910

[13] Chi C C, Alvarez-Mesa M, Juurlink B, et al. Parallel scalability and efficiency of HEVC parallelization approaches[J]. *IEEE Transactions on Circuits & Systems for Video Technology*, 2013, 22(12):1827-1838

[14] Zhou X, Shi G, Zhou W. Perceptual CU size decision and fast prediction mode decision algorithm for HEVC intra coding[C]. In: Proceedings of the 2016 IEEE International Symposium on Multimedia (ISM), San Jose, USA, 2016. 375-378

Zhu Yun, born in 1981. She received her B. S. and M. S. degrees in Microelectronics and Solid State Electronics from Lanzhou University in 2003 and 2006 respectively. She joined the faculty of Xi'an University of Posts & Telecommunications. Since 2014, she has been working on the Ph. D. degree at Xidian University. Her research interests are in the field of array processors.