FPGA Oriented Intra Angular Prediction Image Generation Hardware for HEVC Video Coding

Eita KOBAYASHI†, Seiya SHIBATA†, Noriaki SUZUKI†, Atsufumi SHIBAYAMA†,and Takeo HOSOMI†

† Green Platform Research Laboratories, NEC Corporation,
1753 Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa 211-8666, JAPAN.
{e-kobayashi@fg, s-shibata@ax, n-suzuki@ha, hosomi@ah}.jp.nec.com

Abstract - This work proposes a novel architecture for intra prediction image generation of High Efficiency Video Coding (HEVC) standards oriented to FPGA. HEVC intra prediction is highly-extended from H.264 in those of mode and block size to realize the high flexibility. From the point of view of hardware, however, this flexibility cause an increasing required the number of MUXs although MUXs tend to be a bottleneck of area and frequency in the case of FPGA. In this paper we propose a Reshaping Buffered Architecture which enables reduction the number of MUXs, drastically. Experimental results show that our proposed architecture can reduce up to 70% of number of MUXs compared with conventional raster scan based architecture. This resulted in a marked improvement of maximum frequency by 43% and LUT usage by 51%, respectively.

I. Introduction

Against the growth of mobile terminal, rapid progress of mobile network such as a Long Term Evolution (LTE) and popularization of video on demand (VOD) services, information of video become to be made up of a majority of all of internet traffic in recent years. According to the Cisco Systems report, video IP will be 80% of all of IP traffic by 2018[1]. On the other hands, 4K commercial broadcasting, which has 4 times higher resolution than the Full HD, will come to practical use in next few years. Additionally, 8K Ultra High Definition Television (UltraHD) broadcasting also will be planned to start at 2020 in Japan which has a 16 times higher definition than the current broadcasting. With the context of above situation, High Efficiency Video Coding (HEVC) was standardized at January in 2013 by joint collaboration team of ITU-T and ISO/IEC[2][3]. Because the advantage of the HEVC which can achieve twice as many the compression efficiency as conventional H.264, it is expected as a means of solution of lack of bandwidth in many ways such as commercial broadcasting, mobile traffic and internet.

The algorithm of angular intra prediction image generation of HEVC is largely extended to 33 modes compared with the conventional eight modes in H.264[4]. Prediction block size is also extended to have a choice of those sizes in 4x4, 8x8, 16x16 or 32x32. Those of algorithmic properties of flexibility contribute the improvement both of image quality and compression efficiency[4][5], however the flexibility can be a cause of increasing of amount of calculation.

Although the main issues are area and power consumption in many works concentrating on the hardware architecture for HEVC[6-10], those of large amount of calculation in HEVC drastically increases circuit area and power consumption. In recent years, many research groups target at the Field Programmable Gate Array (FPGA) in place of Application Specific IC (ASIC) because of the rising cost for development ASICs. In the case of FPGA, the inefficiency of multiplexer (MUX) is the crucial problem rather than the case of ASIC. The MUX is a serious bottleneck for both of area and latency in the case of FPGA[1]. Although FPGA has a those deficit, intra prediction image generation module often result in the sea of MUX because of the extended flexibility of those prediction modes and block sizes. As a consequence of this problem, intra prediction image generator tends to have a large area and be a bottleneck of maximum frequency in total HEVC coding hardware.

In this work, we propose a novel architecture to overcome above problems. The essence of our idea is simultaneous generation of prediction pixels by using minimum number of MUXs even if the pixel order will be invalid, and introduction of re-shaping buffer for the sake of consistency. Our architecture we call Reshaping Buffered Architecture is equipped with following two features:

1. Prediction pixels derived from same reference neighbor pixels are generated simultaneously even if some part of prediction pixels are not required to form the valid prediction image. Because generated pixels do not require difference reference pixels individually, only two MUXs are required to generate maximally 32 of prediction pixels. This makes a large contribution to decrease the number of MUX to select the reference neighbor pixels.

2. Prediction pixels are stored to reshaping buffers, and then valid prediction image are read from those buffers by controlling the read and write access addresses. This reshaping buffer can reorder to valid prediction image by using Block RAM (BRAM) even if prediction pixels are generated in various orders. Therefore, any of MUXs are not required to reshaping for the valid prediction image by adapting reshaping buffers. Because some of current FPGAs are equipped with many pre-implemented BRAMs, we think it is better to take advantage of idle BRAMs to reduce the LUT usage unless number of BRAMs is excessive.

The rest of this paper is organized as follows: Section II provides feature of HEVC intra prediction and related works. Section III and Section IV show a detailed behavior of our reshaping buffered architecture and detailed architecture. Then, the synthesis result and comparison will be given in Section V. And finally we draw conclusion in Section VI.

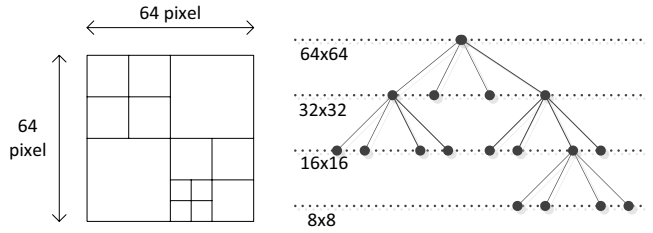

Fig. 1 Dividing pattern of 64x64 LCU into child CUs and quad tree structure corresponding to that dividing.

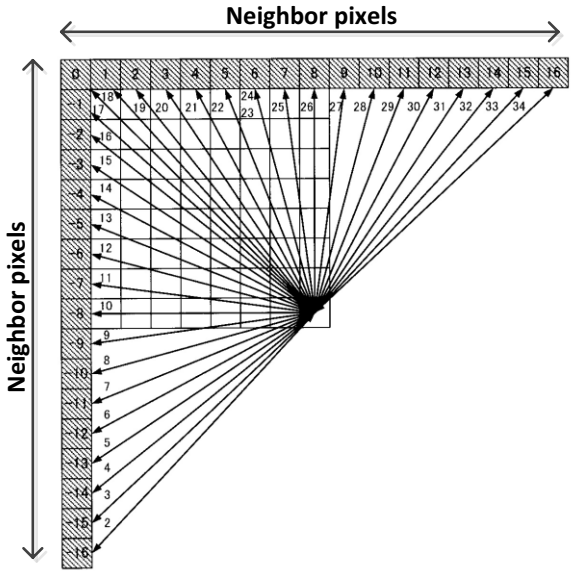

Fig. 2 33 of intra angular mode

II. Preliminaries

A. About HEVC intra prediction

One of the features of HEVC is multiple size of coding unit (CU) and transform unit (TU). A largest CU (LCU) which size is 64x64 pixels are recursively divided into four of squares to form the child CUs. The variety of CU size is 64x64 (LCU), 32x32, 16x16, and minimum size is 8x8. Fig. 1 shows the example of split pattern of LCU with the corresponding quad tree structure. Intra prediction images are generated by unit of TU. TU is also formed recursive split from CU as a root, and TU size ranges from 32x32 to 4x4. This flexibility of block size is one of the significant extended features of HEVC from the H.264/AVC. CU is split into fine grained TUs in the case that part of CU includes complicated figures.

Intra prediction image generation is a making process of prediction image from reference neighbor pixels focused on the spatial continuity of nature image. Intra prediction itself is also improved in respect to high accuracy of prediction angle. Fig. 2 shows 33 of intra angular modes defined in HEVC standard although only eight modes are defined in H.264. The combinations of variety of TU sizes and intra prediction modes contribute to the improvement of image quality and compression efficiency.

B. Related works

In this section we introduce related works of intra prediction.

VLSI architecture for 4x4 TU is developed in [8] for intra prediction in HEVC. Their technique reduces the number of register for holding the reference neighbor pixels. However their technique does not reduce the number of MUXs to select required reference pixels, and their hardware only processes 4x4 TU.

Intra prediction architecture is proposed for all modes and TU size in [9]. However, they did not focus on the internal data path of intra prediction image generation itself. .

Another architecture proposed in [10] support video with 1080p@60fps, but it require quite a large area for two dimensional processing element array. Additionally target of above three works is ASIC.

The work in [11] implemented intra prediction on FPGA concentrate of low power consumption. However, their hardware includes some big MUXs, so we point out that those MUXs are main issue for frequency and LUT usage in the case of FPGA.

Common problem of above works is that they use many large MUXs without hesitating. Our key challenge is to reduce the number of MUX as much as possible.

III. Main Idea

In this section, we explain the problem of this work and propose the solution to overcome those difficulties. As we mentioned before, although the MUX causes the bottleneck of FPGA, it is the fundamental problem that intra prediction image generation module tend to require a lot of MUXs caused by flexibility of algorithm. To eliminate MUXs, we introduce following two techniques into our architecture.

A. Simultaneous prediction image generation

HEVC standard defines that reference pixels are selected in accordance with the pixel coordinate in the prediction block. The value of the prediction samples predSamples[x][y] , with x, y = 0..BlockSize-1 are derived as follows:

$$iIdx = ((y+1)*intraPredAngle) >> 5 \quad (1)$$

$$iFact = ((y+1)*intraPredAngle)\&31 \quad (2)$$

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

where intraPredAngle is specific number associated with each mode, and ref[] is the reference pixel array partially copied form neighbor pixels. Fig. 3 shows reference pixel pair in each position in the 4x4 block at the mode 20 calculated in according to equation (1) and (3). In this figure, (x1, x2) represents reference pixel index pair of above reference pixel array. As shown in this figure, there are some groups of pixels which refer the same reference pixel pair.

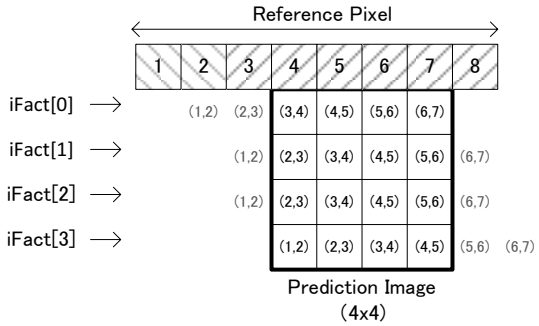

Fig. 3 Group of prediction pixels derived from same reference neighbor pixels.

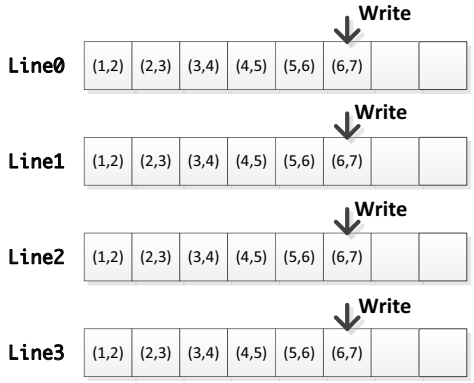

Fig. 4 Condition of the reshaping buffer after the write process.

Our first key idea is simultaneously generating prediction pixels which refer same pixel pair. For example, labeled (3, 4) pixels are generated at a same time by referring the reference pixels indexed 3 and 4 with the iFact multiplication coefficient corresponding to each line derived from equation (2). To avoid the increasing of MUX for the exceptional handling, our method generates all of prediction pixels regardless of whether those pixels are actually included in target square blocks or not. Only two MUXs are required to generate multiple prediction pixels under any modes or block sizes by introducing this idea. In this example, six number of pixel groups labeled (1, 2) to (6, 7), which are consisted of four pixels, are generate respectively.

B. Reshaping buffer and address control

The order of generated pixel groups by the strategy as mentioned in section III.A is neither raster scan order nor square block unit, additionally it contains redundant pixels. Because there are many patterns of generated prediction pixels order vary with 33 of modes and four of block sizes, later module has to be equipped with the function of reshaping those pixels in a constant valid order and cutting of redundant pixels.

Fig. 4 shows a condition of reshaping buffer after the write process. Simultaneously generated pixel group is written in a same address of each memory corresponding to each line. Note that only one pixel is generated per one line in the regular process of HEVC standards.

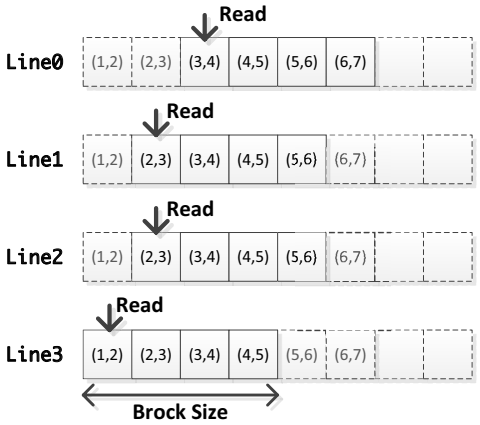

Fig. 5 Read process from the reshaping buffer.

Fig. 5 shows a read address at the beginning of read process. Each of start address of each line is different in accordance with the given mode and block size. Block size also represent the number of read times. Start from given start address of each line, prediction pixels are read column by column of TU by increment the read address. In this process, redundant prediction pixels are never read from buffers, in addition, column by column order is also guaranteed without any of MUXs.

IV. Proposed Architecture

A. Overview

Our intra prediction image generator is consisted of three modules. Each module is NEI_COPY, IPRED, and BUF_CTRL. Fig.6 shows a schematic of our architecture and interfaces between those modules. Those three modules are connected sequentially and compose the macro pipeline architecture. The prediction image in TU is generated by an iteration of those series of pipeline process. When the former modules ended the process, valid (or fin) signal and required parameters, which are mode and block size, are transmitted simultaneously to the later module. The later module gets the parameters when receives the valid signal and starts to process. When the FIN signal is transmitted from the final module (BUF_CTRL) transmits to the initial module (NEI_COPY), the next prediction image generation process is driven in next TU. Table1 and Table2 show inputs and outputs of our system specifications.

B. Partial copy of neighbor pixels (NEI_COPY)

NEI_COPY provides the projection process from left reference pixels to the extended above reference pixel array under the regulation of HEVC standards as shown in Fig. 7. The reference pixel array ref[x] is specified as follows:

$$ref[x] = p[-1+((x*invAngle + 128) >> 8)] \quad (4)$$

where invAngle is specific number associated with each mode, and p[] is the left reference pixel array. Fig.5 shows a function of NEI_COPY.

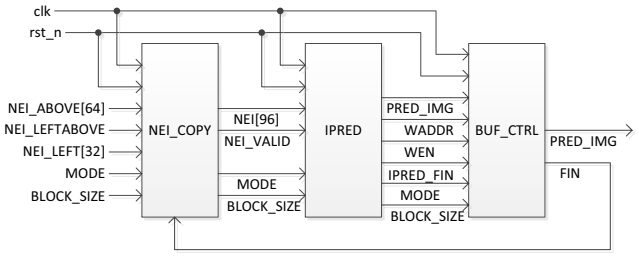

Fig. 6 Total pipelined architectures of our intra prediction image generator. Child modules are NEI_COPY, IPRED, and BUF_CTRL.

32 of left reference pixels from NEI_LEFT[0] to NEI_LEFT[31] are candidates which are copied to extended above reference pixel array. This projection is executed in parallel under the assumption that block size is 32x32 regardless of actual block size to avoid the increasing of number of MUXs. When finish to project, NEI_COPY module will send the extended above reference pixel array to the next module with the valid signal, mode and block size.

C. Part of intra prediction image generation (IPRED)

Simultaneous prediction pixels generation mentioned in section III.A is realized as IPRED module. Fig. 8 shows a detailed architecture of IPRED. This module is equipped with three features; reference pixel buffer, selector, and calculator for the value of prediction pixel. Reference pixel buffer preserves extended above reference pixels given from the NEI_COPY.

Selector is the 96 to 1 MUX which choices two of reference pixels from 96 of reference pixel buffer. Selected two pixels are provided 32 of Processing Elements (PE) which calculate values of prediction pixels based on the equation (3) in section III.A. iFact[] are also calculated based on the equation (2) by using given block size. Reading start index of reference pixel buffer and number of iteration, which differ depending on the mode and block size, are calculated in advance and preserved in ROMs.

Those ROMs are referenced by using mode and block size as index when the valid signal is given from former module.

Table 1 Input port definition

Name	Description
NEI_ABOVE[64]	8bit x 32 array of above neighbor pixels
NEI_LEFTABOVE	8bit of left above neighbor pixel
NEI_LEFT[32]	8bit x 32 array of left neighbor pixels
MODE	6bit of specified mode of this TU
BLOCK_SIZE	Block size represented as 2bit. 2b'00:4x4 2b'01:8x8 2b'10:16x16 2b'11:32x32

Table 2 Output port definition

Name	Description
PRED_IMG[32]	8bit x 32 array of prediction image.

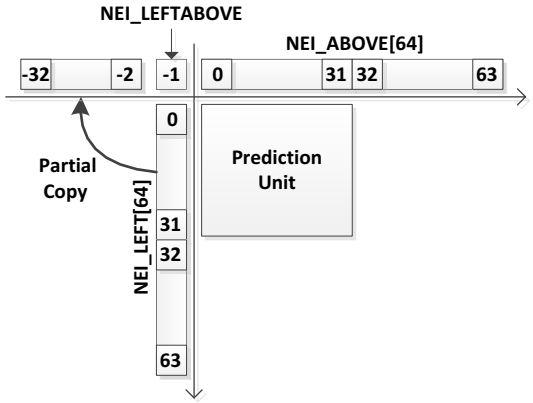

Fig. 7 Partial copy process in NEI_COPY

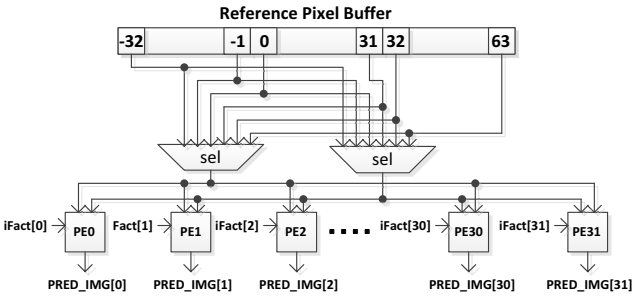

Fig. 8 Architecture of IPRED module

D. Buffer control (BUF_CTRL)

BUF_CTRL includes reshaping buffer and buffer controller as mentioned in section III.B. Fig. 9 shows a detailed architecture of BUF_CTRL. Reshaping buffer includes 32 of line buffer because maximum block size is 32x32. Input signals, a write enable (WEN), a prediction pixels (PRED_IMG) and a write address (WADDR) are directory connected to the reshaping buffer. When fin signal (IPRED_FIN) is given to this module from IPRED, buffer controller gets the mode and block size at a same time, and then the read process are started. Buffer controller refer the ROM tables by using mode and block size to get the start address of each line buffer in reshaping buffer. FIN signal is generated at the end of read process from buffer controller after all of prediction pixels are generated.

V. Experimental Result

The proposed architecture is designed by systemC and implemented in a Xilinx viretex-7 series FPGA which is equipped with six inputs and two output LUTs. We use CyberWorkBench as high-level synthesis tool and Vivado 2014.3 as logic synthesis and implementation tool. In this work, target frequency is 100MHz although the target frequency can be set flexibly by using high-level synthesis tool with untimed systemC model. We evaluate the final implementation result through the design flow starts from systemC. We also verified the correctness of generated RTL by using Modelsim as a RTL simulator.

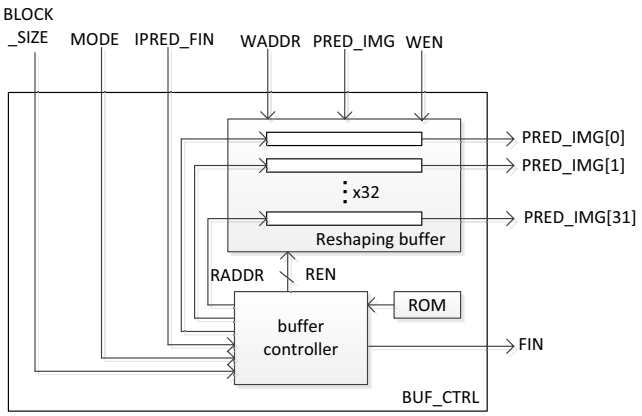

Fig. 9 Architecture of BUF_CTRL module.

Table 3 shows the result of latency to generate the valid NxN square prediction image. Note that IPRED is synthesized as two stage pipelined architecture by high-level automatic loop-folding synthesis. In this work, indeed, there is a room for improvement of latency. Read process could be started when the all of valid prediction pixels are generated in first column of TU. There is no need to wait until all of prediction pixels in TU are generated.

Table 4 shows a resource usage of each module and total modules. As shown this table, the biggest module is NEI_COPY because it includes many MUXs to select the pixel to be copied and lead to the adequate position of extended above reference pixel array. Note that F7 MUX and F8 MUX are embedded multiplexers in FPGA as shown in Fig.10. NEI_COPY became relatively large modules compared with IPRED and BUF_CTRL derived from our proposed reshaping buffered architecture. It remains a matter of research for efficient FPGA oriented NEI_COPY architecture. Table 5 shows a resource usage comparison of our reshaping buffered architecture and conventional raster scan architecture, which corresponds to set of IPRED and BUF_CTRL in our proposed architecture. NEI_COPY is omitted in this evaluation because NEI_COPY is commonly required function both of our proposed architecture and raster scan architecture, and is a still subject for a further study. Although this raster scan architecture, as shown in the Fig. 11, also implemented in pipelined architecture with parallel PEs like our proposed architecture, there is a difference in that it can generate one line of prediction pixels at a same time. While the raster scan architecture does not require the reshaping process to generate in a valid order, it has to require many MUXs to refer the different reference pixels individually. As shown in Table 5, our proposed architecture can reduced usage of Slice LUT, F7 MUX and F8 MUX up to 70% compared with the raster scan architecture. We don't think the increasing of register usage as a significant problem because there is much room for register usage ratio in most of FPGA-targeted developing situation, especially in the case of signal processing. Additionally, number of BRAMs is reasonable compared to LUT usage.

Table 3 Total number of required cycles to generate the NxN prediction images.

	NEI_COPY	IPRED	BUF_CTRL	total
4x4	2	5	4	11
8x8	2	9	8	19
16x16	2	17	16	35
32x32	2	33	32	67

Table 4 Resource usage of each module in our proposed architecture

	NEI_COPY	IPRED	BUF_CTRL	total
Slice LUTs	3381	1534	376	5297
Slice Registers	1806	1202	1306	4314
F7 MUX	756	97	0	853
F8 MUX	64	48	0	112
BRAM	0	0	16	16
DSP48	0	64	0	64

Table 6 shows a maximum frequency comparison of our reshaping buffered architecture and raster scan architecture. We estimate the maximum frequencies of two architectures by using Synopsys Simplify_pro which can estimate maximum frequency from RTL. We confirm that our proposed architecture including three modules achieves 43% of improvement by the contribution of degreasing the number of MUXs.

Those advantages of our architecture could be achieved owing to embedded BRAMs and disadvantage of MUXs in FPGA. We think increasing number of BRAMs could be evaluated as reasonable because it is compatible the number of LUTs usage.

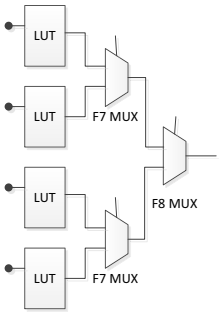

Fig. 10 F7 and F8 MUX in FPGA

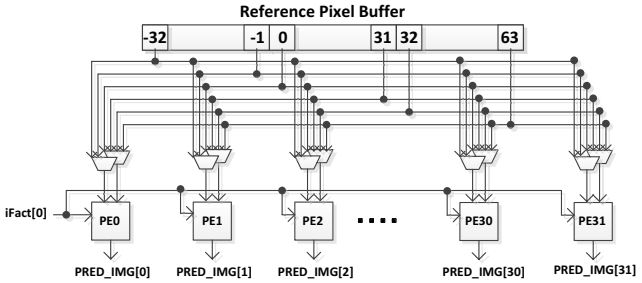

Fig. 11 Raster scan based architecture

Table 5 Comparison the resource usage of our proposed architecture and raster scan based architecture.

	Raster scan architecture	IPRED + BUF_CTRL	Ratio
Slice LUTs	3693	1910	-51%
Slice Registers	1211	2508	+100%
F7 MUX	329	97	-71%
F8 MUX	96	48	-50%
BRAM	0	16	N/A
DSP48	64	64	0%

Table 6 Comparison the maximum frequency of our proposed architecture and raster scan based architecture.

	Raster scan architecture	Proposed architecture	Ratio
Maximum Frequency.	104.3 MHz	149.2 MHz	+43%

VI. Summary and Conclusions

In this work, we proposed FPGA oriented reshaping buffered architecture. Because MUX is the bottleneck of FPGA, we introduce combination of two types of techniques to reduce the number of MUXs: simultaneous generation of prediction pixels which refer same reference pixels and reshaping buffer by using one-dimensional series of memories. Our proposal achieved 51% of reducing the LUT usage and 43% of improvement in maximum frequency compared with raster scan based architecture without our proposed techniques. Those advantages could contribute to realize FPGA based HEVC coding.

References

[1] Cisco white paper, “Cisco Visual Networking Index: Forecast and Methodology, 2013-2018,” Feb. 2014.

[2] ITU-T H.265|ISO/IEC 23008-2 High efficiency video coding

[3] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Trans. Circuits and Systems for Video Technology, Vol. 22, No. 12, pp. 1649-1668, Dec

[4] Lainema, J.; Bossen, F.; Woo-Jin Han; Junghye Min; Ugur, K., "Intra Coding of the HEVC Standard," Circuits and Systems for Video Technology, IEEE Transactions on , vol.22, no.12, pp.1792,1801, Dec. 2012.

[5] W.-J. Han et al.: "Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools," IEEE Trans. Circuits Syst. Video Technol., 20(12), pp. 1709-1720, Dec. 2010.

[6] C. Huang, M. Tikekar, C. Juvekar, V. Sze, and A. Chandrakasan, "A 249Mpixel/s HEVC video-decoder chip for Quad Full HD applications", ;in Proc. ISSCC, 2013, pp.162-163.

[7] Jinjia Zhou, Dajiang Zhou, Wei Fei, and Satoshi Goto,” A High-performance CABAC Encoder Architecture for HEVC and H.264/AVC”, International Conference on Image Processing(ICIP), Melbourne, Australia, pp. 1568-1572 ,Sept, 2013

[8] Li, Fu, Guangming Shi, and Feng Wu. "An efficient VLSI architecture for 4× 4 intra prediction in the High Efficiency Video Coding (HEVC) standard." Image Processing (ICIP), 2011 18th IEEE International Conference on. IEEE, 2011.

[9] Palomino, Daniel, et al. "A memory aware and multiplierless VLSI architecture for the complete Intra Prediction of the HEVC emerging standard." Image Processing (ICIP), 2012 19th IEEE International Conference on. IEEE, 2012.

[10] Ning Zhou, Dandan Ding, Lu Yu. “On hardware architecture and processing order of hevc intra prediction module.” 30th Picture Coding Symposium (PCS), 2013

[11] Kalali, E.; Adibelli, Y.; Hamzaoglu, I., "A high performance and low energy intra prediction hardware for High Efficiency Video Coding," Field Programmable Logic and Applications (FPL), 2012 22nd International Conference on , vol., no., pp.719,722, 29-31 Aug. 2012

[12] Metzgen, P.; Nancekievill, D., "Multiplexer restructuring for FPGA implementation cost reduction," Design Automation Conference, 2005. Proceedings. 42nd , vol., no., pp.421,426, 13-17 June 2005