A Low-Cost Complexity Intra Prediction Hardware Architecture for HEVC Decoder

Hongkyun Jung, Kwangki Ryoo

Abstract: ***High Efficiency Video Coding (HEVC) adopts new techniques to reduce bit-rate by 50% over a previous video compression standard. The number of intra prediction modes in HEVC is 35 modes and increased compared with the compression. Therefore, hardware architecture with common equation and a fast filter coefficient generation algorithm is proposed for low complexity intra prediction hardware. The proposed architecture performs a smoothing filter, interpolation filter, generation of predicted pixels with only Common Operation Unit (COU). Various equations in intra prediction for smoothing filter of reference samples, calculating the average of the reference samples, generating predicted pixels and filtering predicted pixels is modified to one common equation. The common operation unit using a common equation in intra prediction hardware architecture reduces hardware area and the number of computational operators to perform various equations. COU uses 2 multipliers, 9 adders, 3 shifters and generates 1 predicted sample in planar mode and 2 predicted samples in the other mode. Also, COU generates 2 filtered reference samples in filtering operation of reference samples and the average of 4x4 PU in DC mode. The fast filter coefficient generation algorithm reduces processing time by using only Look-Up Table (LUT) and adders, instead of multiplying operation and the number of computational operators. The number of gates of the architecture is 45.6k. The number of gates in the proposed intra prediction hardware is 36.7% less than previous architecture.***

Keywords : HEVC, Intra Prediction, Common Operation Unit, Fast Filter Coefficient Generation, Hardware Architecture.

I. INTRODUCTION

HEVC is developed as the most recent video compression standard of the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group [1]. The standard adopts new techniques to reduce bit-rate by 50% over a previous video compression standard. In intra prediction, up to 35 prediction modes, including DC, planar, and 33 angular modes, are supported and the number of prediction mode is increased compared with 9 prediction modes in H.264/AVC [2, 3].

In intra prediction, various sizes of Prediction Unit (PU) from 4x4 PU to 32x32 PU are supported to generate the prediction picture most similar to the current picture [4]. As the intra prediction of HEVC supports 35 prediction modes and various sizes of PU and processes the reference pixel smoothing process and the filtering process of the predicted pixel, the amount of computations and the computational complexity are greatly increased, compared to the existing H.264/AVC in implementation of the hardware [5]. Several architectures for hardware implementation of intra prediction in HEVC have been discussed in [6-9]. The architecture in [6] proposed a flexible reference sample selection technique and a register array to reduce memory resource. The architecture in [6] adopts an efficient uniform architecture for Angular modes and DC mode. The design in [7] adopts two pipelines to produce 4x4 pixel data with varied throughputs and a method to process reference pixel fetching and padding. The design in [8] proposed an efficient VLSI architecture to support 8Kx4K HEVC decoder and a cyclic SRAM banks based parallel reference sample fetching. The design consists of common operation unit for generating predicted pixels for all modes and reference pixel substitution unit to process reference substitution and smoothing. The designs in [6-8] implemented two or three operation units to generate predicted samples. Also, designs in [7-10] implemented a filtering operation unit for reference pixels.

Therefore, low complexity intra prediction hardware architecture with COU and a fast filter coefficient generation algorithm is proposed to reduce execution cycles of PU and computational units for HEVC decoder.

The proposed architecture uses common operation unit applying a common equation that computes smoothing, interpolation, predicted pixel generation and the average in DC mode. The algorithm uses only LUT and adders instead of multipliers and logical and operator. Therefore, the algorithm can reduce hardware area and path delay.

II. INTRA PREDICTION IN HEVC

In the intra prediction process of HEVC, a reference pixel smoothing process performs 3-tap filtering operation for the current reference pixel, left and right reference pixels to reduce the error of the prediction. Whether or not the process is performed is decided by the size of PU and the prediction mode.

Equation (1) is applied with 3 tap filter for smoothing of the reference pixels located on the upper of the current PU. pF is a smoothed reference sample, p is an unsmoothed reference sample. y means the row location of the reference sample, the coefficient of the current reference sample is 1/2, and the coefficient of the left/right pixels is 1/4.

$$pF[-1, y] = (p[-1, y+1] + 2 * p[-1, y] + p[-1, y-1] + 2) >> 2$$

$$\text{for } y = nS * 2-2 \text{ to } 0 \qquad (1)$$

Revised Manuscript Received on July 22, 2019.

Hongkyun Jung, Large Touch Team, Silicon Works, Seoul, South Korea. Email: junghk@siliconworks.co.kr

Kwangki Ryoo*, Department of Information and Communication Engineering, Hanbat National University, Daejeon, South Korea. Email: kkryoo@gmail.com

In the prediction, the predicted sample is generated by applying the predicted sample generation equation corresponding to each prediction mode and is filtered by applying the interpolation filter to the first column and the first row of PU in DC mode, to the first column in Horizontal mode, and to the first row in Vertical mode.

Equation (2) is for calculating the average of the reference samples in DC mode. DCVal is for the average of the reference samples, p for reference samples, nS for the size of PU, x' for the location of the column, and y' for the location of the row.

$$DCVal = (\sum p[x', -1] + \sum p[-1, y'] + nS) \gg (k+1) \text{ for } x = nS-1 \text{ to } 0,\ y = nS-1 \text{ to } 0,\ k = \log 2nS \quad (2)$$

Apart from Horizontal and Vertical modes, (3) to (5) are the ones that is used to generate the predicted samples in Angular mode. In (3), iFact, filter coefficient is decided according to interPredAngle, logic AND operation and y, indicating the location of the reference samples.

$$iFact = ((y+1) * \int raPredAngle) \,\&\&\, 31 \quad (3)$$

Equation (4) generates iIdx, indicating the location of the reference pixels, and iIdx is decided according to intraPredAngle, y and shift operation.

$$iIdx = ((y+1) * \int raPredAngle) >> 5 \quad (4)$$

Equation (5) generates the predicted pixels by applying the interpolation filter to 2 reference pixels selected by the directional information of the prediction mode.

$$pS[x, y] = ((32 - iFact) * (r[x+iIdx+1] + iFact * r[x+iIdx+2] + 16) >> 5 \quad (5)$$

III. THE PROPOSED HARDWARE ARCHITECTURE

As in Fig. 1, the proposed architecture consists of internal buffers, memory controller, mode generator (ModeGen), reference pixel padding unit (REFPIXPAD), filter decision unit (FilterDecision), parameter generator (iFact & Coef Gen) and 16 common operation units (COU). ModeGen generates prediction mode of current PU and FilterDecision determines whether a 3-tap filter operation for reference samples is performed. COU generates the average value in DC mode, predicted samples and performs smoothing process for reference samples.

Filter decision unit adopts a fast smoothing decision algorithm that decides the smoothing process only with comparison of the bits of the prediction mode without arithmetic operation, instead of the smooth decision algorithm. Internal buffers are stored average value in DC mode, padded reference pixels and filtered reference samples.

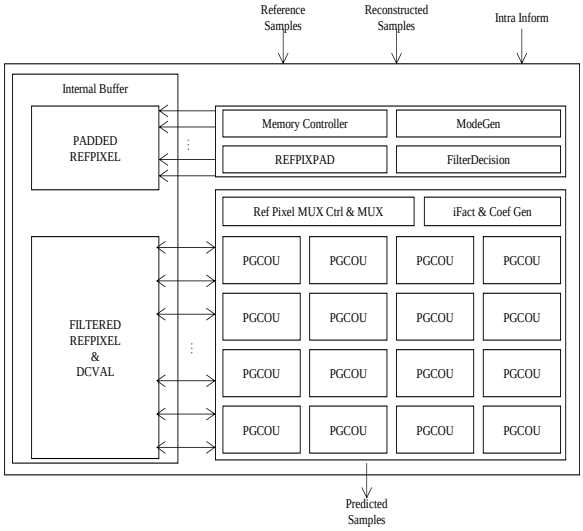

Fig. 1. The Proposed Hardware Architecture

A. Fast Filter Coefficient Generation Algorithm

The proposed fast filter coefficient generation algorithm generates the filter coefficient in angular mode with only LUT, instead of adder, multiplier and logic AND operation applied with Equation (3). iFact, which has been calculated by applying with equation (3) for all the prediction mode and y corresponding to Angular mode, is shown as Table I and Table II. iFact for each y resulted from taking the lower 5 bits after multiplying y+1 by iFact, when y is 0. For example, in the case that the prediction mode is 4, iFact is 11 when y is 0, and iFact is 22 when y is 1. As the lower 5 bits is taken from 33 that is a result of multiplying between y+1 and iFact, when y is 0, in the case that y is 2. iFact for the prediction mode is horizontally symmetrical in the center of the prediction mode 18.

Table- I: iFact for Angular Modes 2-18

y / mode	0	1	2	3	4	⋯	30	31
2	0	0	0	0	0	⋯	0	0
3	6	12	18	24	30	⋯	26	0
4	11	22	1	12	23	⋯	21	0
5	15	30	13	28	11	⋯	17	0
6	19	6	25	12	31	⋯	13	0
7	23	14	5	28	19	⋯	9	0
8	27	22	17	12	7	⋯	5	0
9	30	28	26	24	22	⋯	2	0
10	1	2	3	4	5	⋯	31	0
11	2	4	6	8	10	⋯	30	0
12	5	10	15	20	25	⋯	27	0
13	9	18	27	4	13	⋯	23	0
14	13	26	7	20	1	⋯	19	0
15	17	2	19	4	21	⋯	15	0
16	21	10	31	20	9	⋯	11	0
17	26	20	14	8	2	⋯	6	0
18	0	0	0	0	0	⋯	0	0

Table- II: iFact for Angular Modes 19-34

y / mode	0	1	2	3	4	⋯	30	31
19	26	20	14	8	2	⋯	6	0
20	21	10	31	20	9	⋯	11	0
21	21	10	31	20	9	⋯	11	0
22	17	2	19	4	21	⋯	15	0
23	13	26	7	20	1	⋯	19	0
24	9	18	27	4	13	⋯	23	0
25	2	4	6	8	10	⋯	30	0
26	1	2	3	4	5	⋯	31	0
27	30	28	26	24	22	⋯	2	0
28	27	22	17	12	7	⋯	5	0
29	23	14	5	28	19	⋯	9	0
30	19	6	25	12	31	⋯	13	0
31	15	30	13	28	11	⋯	17	0
32	11	22	1	12	23	⋯	21	0
33	6	12	18	24	30	⋯	26	0
34	0	0	0	0	0	⋯	0	0

If applying the mode-2 mode resulting from deducting 2 from the prediction mode in Table I and Table II, as iFact from the mode-2 mode 0 to the mode-2 mode 15 and iFact from the mode-2 mode 17 to the mode-2 mode 32 become horizontally symmetrical, and the result of deducting the mode-2 mode 17 to 32 from 32 becomes the mode-2 from 15 to 0, the Table I and Table II are simplified to iFact of the mode-2 mode 0 to 16. An addition between the iFact from the prediction mode 2 to the prediction mode 9 of the iFact in Table I and Table II and the iFact from the prediction mode 18 to the prediction mode 11, respectively, become 32. Therefore, an operation subtracting iFact of the prediction mode 2 to 9 from 32 is resulted in iFact of the prediction mode 18 to 11, Table I and Table II are simplified to iFact of the prediction mode 2 to 9.

If iFact necessary for 4x4 PU after applying the mode-2 mode and the simplification technique to Table I and Table II 6 is summarized, Table I and Table II can be applied as a simple LUT, shown as Table III. The offset in Table 3 is used for generating iFact necessary for 8x8 PU, 16x16 PU, and 32x32 PU.

The fast filter coefficient generation hardware architecture applied with the simplified Table 3 and the adding operation, as in Fig. 2.

Table- III: Simplified iFact

Mode-2	iFact0	iFact1	iFact2	iFact3	offset
0	0	0	0	0	0
1	6	12	18	24	24
2	11	22	1	12	12
3	15	30	13	28	28
4	19	6	25	12	12
5	23	14	5	28	28
6	27	22	17	12	12
7	30	28	26	24	24

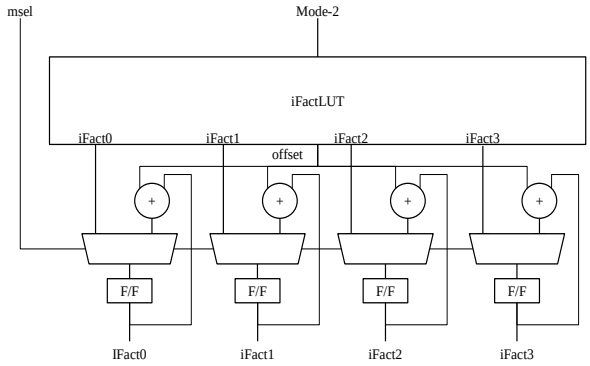

Fig 2. Fast Filter Hardware Architecture

The architecture consists of iFactLUT, 4 adders, and registers. iFactLUT is saving iFact of Table 3, and output 4 iFacts and offsets corresponding to the mode –2 mode, once the mode-2 has been input. In case of processing 4x4 PU and the first 4x4 block of 8x8 PU, 16x16 PU and 32x32 PU, 4 iFacts from LUT is output and 4 iFacts corresponding to the location of each y is output by adding 4 iFacts and offsets saved in the register from the second 4x4 block of 8x8 PU, 16x16 PU, and 32x32 PU.

B. Proposed COU

The proposed COU uses the common equation applying modified equations generating predicted samples in prediction modes and an equation computing the average value in DC mode. Equation (6) is common operation equation. the equation is applied in all filtering process and generating process of predicted pixels.

$$p[x,y] = (((((A0+A1+1)*D0 + (C0+C1)) >> Len) + B0) + ((((A2+ A3+1)*D1 + (C2+C3)) >> Len) + B1) +1) >>1 \quad (6)$$

Equation (7) is (1) applied the common equation by modifying location of p[-1, y].

$$pF[x,y]=(2 * (1-p[-1,y]) + p[-1,y+1] + p[-1,y-1]) >> 2 + p[-1,y] \quad (7)$$

Equation (8) is (2) for 4X4 PU. Shift operator is separated, and multiplication operator is inserted in equation 2 to applying the common equation.

$$DCVal=(((1*(p[0,-1]+p[1, -1]+1) + (p[2, -1] +p[3, -1] +1)) >>2 + ((1*(p[-1,0]+p[-1,1]+1) + (p[-1,2]+p[-1,3] +1))>>2 +1)>>1 \quad (8)$$

As in Fig. 3, COU uses 9 adders, 3 shifters and 2 multipliers, and generates 1 predicted pixel in planar mode, 2 predicted pixels in other mode, and the average value of 4x4 PU in DC mode and 2 filtered reference pixels in filtering operation of reference samples.

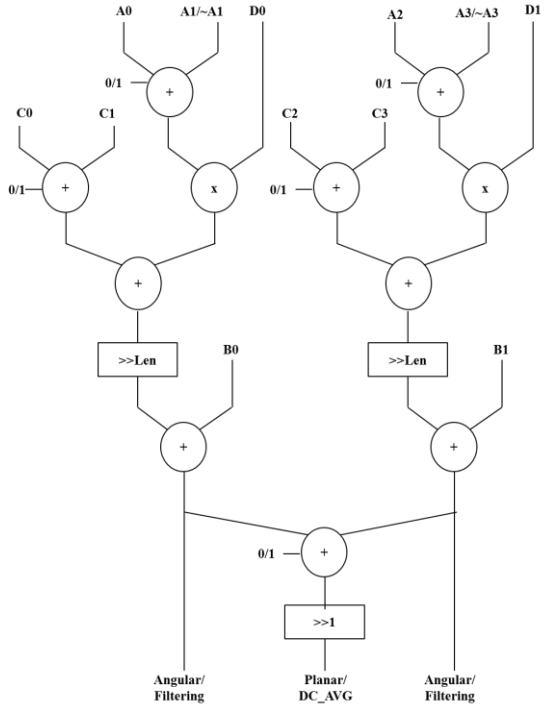

Fig 3. Common Operation Unit

To apply (7), COU uses 8 adders, 2 shifters and 2 multipliers except to 1 adder and 1 shifter in a last stage and generates 2 filtered samples, as in Fig. 4. In DC mode, COU uses all operators and generates the average of reference samples to apply (8), as in Fig. 5

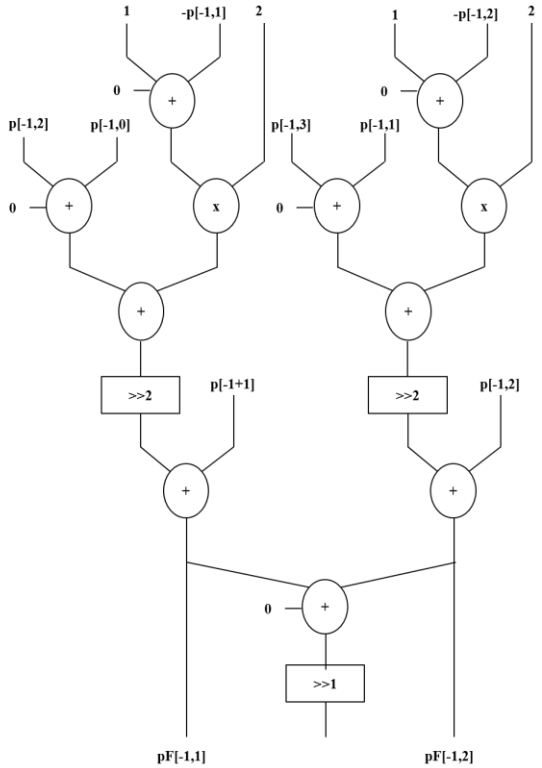

Fig 4 . COU applying 3-tap filter for smooth filtering

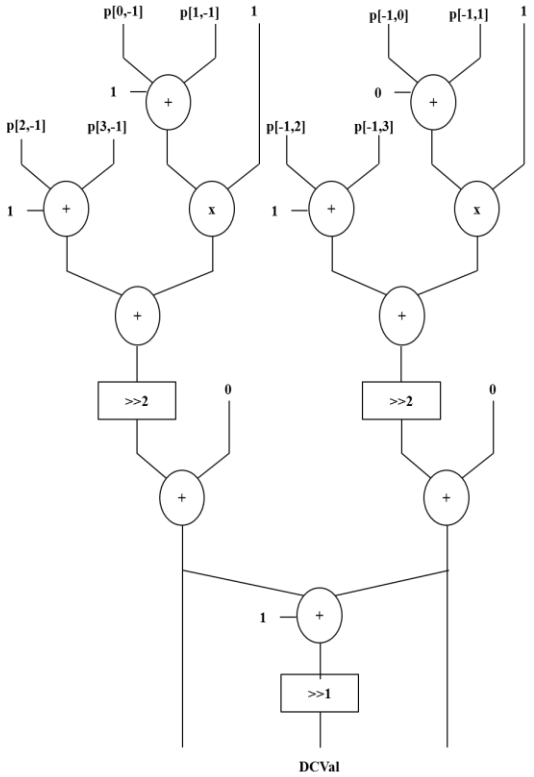

Fig. 5. COU for DCVal in DC mode

IV. EXPERIMENTAL RESULT

The proposed hardware architecture is designed with Verilog HDL and synthesized by the TSMC 0.13um standard cell library. Table V shows the synthesis results of the proposed design and the comparison width previous designs. The gate count in the architecture is 45.6k. The maximum operating frequency of the proposed architecture is 164 MHz.

Table- V: Synthesis Result and Comparison with Previous Designs

	[6]	**[7]**	**[9]**	**Proposed**
Available PU	4x4	4x4-32x32	4x4-32x32	4x4-32x32
Filtering	N/A	Support	Support	Support
Technology (nm)	130	90	90	130
Maximum Frequency (MHz)	150	27	469	164
output pixels	1	2	16	16
Execution Cycles (4X4/8X8/ 16X16/32X32)	16/-/ -/-	9/36/ 144/576	1/4/ 16/128	1/4/ 16/128
Gate count	9k	76.8k	72.1k	45.6k

Compared with the architecture in [7], the proposed architecture reduces the number of execution cycles by 88.8%. The number of gates in the proposed architecture is 36.7% less than the architecture in [9].

As in Fig. 6, HEVC reference software HM 10.0[11] is used to verify the proposed architecture, and test video sequence is encoded by HM 10.0., and then bitstream is generated. The input data and output data of the intra prediction functions have been extracted by HM 10.0 decoder to compare the output data of the proposed architecture. Comparing output data from the proposed architecture with output data from intra prediction functions in HM 10.0 decoder, they match each other.

COU of the proposed architecture uses 10 adders, 2 multiplier and 3 shifter and PE in [9] uses 2 multiplier, 14 adders, 7 multiplexers. The number of adders in COU is about 35.7% less than PE in [9] and COU don't use multiplexer. In addition, as COU supports smoothing filtering, COU is more efficient than PE in [9] as shown in Table IV.

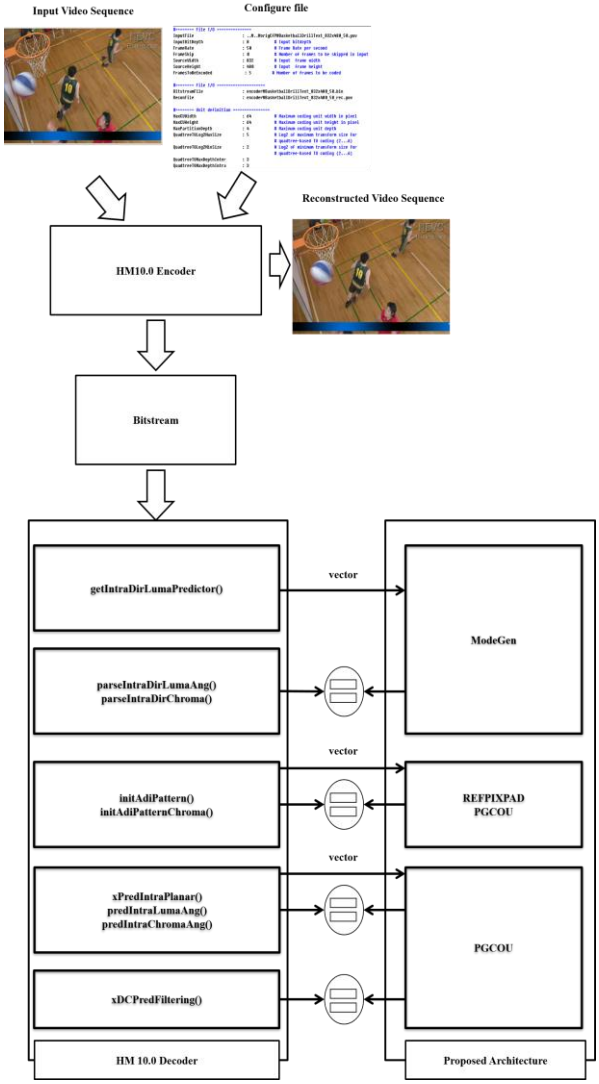

Table- IV: Comparison of Operator Resource in COU

Resource	[9]	Proposed
Adder	14	9
Multiplier	2	2
Shifter	-	3
Multiplexer	7	-

V. CONCLUSION AND FUTURE DIRECTION

Intra prediction in HEVC supports various PU size and 35 prediction modes and Compared with the previous standard, computational operators and hardware area for the prediction are increased. Therefore, low complexity intra prediction hardware architecture with COU and fast filter generation algorithm is proposed to reduce hardware area and computational complexity. COU processes smoothing for reference pixels, filter operation of predicted pixels and generating predicted pixels. The algorithm uses LUT and adders instead of multiplier, adders, shifters. As all filtering and generation of predicted pixels is processed by only COU, computational complexities and hardware area of the architecture are reduced. As the experimental results, the number of gates in intra prediction hardware with the architecture is 36.7% less than previous architecture.

REFERENCES

1. G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, Dec. 2012, pp. 1649–1668.
2. D. Zhou, S. Wang, H. Sun, J. Zhou, 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 Journal of Solid-State Circuits, vol. 52, Jan. 2017, pp. 113–126.
3. T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, Jul. 2003, pp. 560–576.
4. 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 Journal of Solid-State Circuits, vol. 49, Jan. 2014, pp. 61–72.
5. J. Lainema, F. Bossen, W. J. Han, J. M. Min, and K. Ugur, "Intra Coding of the HEVC Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, Dec. 2012, pp. 1792–1801.
6. F. Li, G. Shi, and F. Wu, "An efficient VLSI architecture for 4×4 intra prediction in the High Efficiency Video Coding (HEVC) standard," 18th IEEE International Conference on Image Processing, Brussels, Belgium, Sep. 2011.
7. P. T. Chiang, Y. C. Ting, H. K. Chen, S. Y. Jou, I. W. Chen, and H. C. Fang, "A QFHD 30-frames/s HEVC Decoder Design," IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, Apr. 2016, pp. 724–735.
8. M. Abeydeera, M. Karunaratne, G. Karunaratne, K. D. Silva, and A. Pasqual, "4K Real-Time HEVC Decoder on an FPGA," IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, Jan. 2016, pp. 236–249.
9. J. Zhou, D. Zhou, H. Sun, and S. Goto, "VLSI architecture of HEVC intra prediction for 8K UHDTV applications," IEEE International Conference on Image Processing, Paris, France, Oct. 2014.
10. 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 Transactions on Very Large Scale Integration Systems, vol. 22, Jul. 2014, pp. 1515–1525.
11. HM 7.1 Reference Software, Available: http://hevc.kw.bbc.co.uk/trac/browser/tags/HM-10.0

AUTHORS PROFILE

Hong-Kyun Jung was awarded his BSc., MSc, and Ph.D. Degrees in Electronic Engineering from Habbat National University, Korea in 2007, 2009 and 2019 respectively. Since 2013, he has been a Research Engineer at Silicon works Co. Ltd., Seoul Korea. His research interests include SoC Platform Design, Multimedia Codec Design and Touch System Design.

Kwangki Ryoo was awarded his BSc., MSc, and Ph.D. Degrees in Electronic Engineering from Hanyang University, Korea in 1986, 1988 and 2000 respectively. From 1991 to 1994, he was an Assistant Professor at the Korea Military Academy (KMA) in South Korea. He later worked as a Senior Researcher at the Electronics and Telecommunications Research Institute (ETRI), Korea, from 2000 to 2002. He was a Visiting Scholar at the University of Texas in Dallas from 2010 to 2011. Since 2003, he has been a Professor at Hanbat National University, Daejeon Korea. His research interests include Engineering Education, SoC Platform Design and Verification, Image Signal Processing and Multimedia Codec Design, and SoC Design for Security.