Pixel Correlation Based Computation and Energy Reduction Techniques for HEVC Fractional Interpolation

Ercan Kalali, Ahmet Can Mert, Ilker Hamzaoglu
Faculty of Engineering and Natural Sciences, Sabanci University
34956 Tuzla, Istanbul, Turkey
{ercankalali, ahmetcanmert, hamzaoglu}@sabanciuniv.edu

***Abstract*— Fractional interpolation is one of the most computationally intensive parts of High Efficiency Video Coding (HEVC). Therefore, in this paper, two pixel correlation based computation and energy reduction techniques for HEVC fractional interpolation are proposed. The proposed pixel equality based computation reduction (PECR) technique does not affect the PSNR and bit-rate. The proposed pixel similarity based computation reduction (PSCR) technique slightly decreases PSNR and increases bit-rate. In this paper, a low energy HEVC fractional (half-pixel and quarter-pixel) interpolation hardware for all prediction unit sizes including the proposed techniques is also designed and implemented using Verilog HDL. The proposed hardware, in the worst case, can process 48 quad HD (2160x1600) video frames per second. The proposed PECR and PSCR techniques reduced the energy consumption of this hardware up to 39.7% and 46.9%, respectively.**

Keywords—HEVC, Fractional Interpolation, Hardware Implementation, FPGA, Energy Reduction.

I. INTRODUCTION

A new international video compression standard called High Efficiency Video Coding (HEVC) is recently developed [1]-[6]. It has 50% better video compression efficiency than H.264. In order to increase the performance of integer pixel motion estimation, fractional motion estimation is performed in HEVC. Fractional interpolation is one of the most computationally intensive parts of HEVC video encoder and decoder. On average, one fourth of the HEVC encoder complexity and 50% of the HEVC decoder complexity are caused by fractional interpolation [7].

In H.264 standard, a 6-tap FIR filter is used for half-pixel interpolation and a bilinear filter is used for quarter-pixel interpolation [8]. In HEVC standard, 3 different 8-tap FIR filters are used for both half-pixel and quarter-pixel interpolations. In H.264, 4x4 and 16x16 block sizes are used. However, in HEVC, prediction unit (PU) size can be from 4x4 to 64x64. Therefore, HEVC fractional interpolation is more complex than H.264 fractional interpolation.

Two pixel correlation based computation and energy reduction techniques (pixel equality based computation reduction (PECR) and pixel similarity based computation reduction (PSCR)) are proposed for HEVC intra prediction in [4]. In this paper, these techniques are applied to HEVC fractional interpolation. The proposed techniques compare the pixels at the inputs of HEVC fractional interpolation operation. If these pixels are equal or similar, interpolation operation is skipped and one of the input pixels is selected as output. Therefore, the computational complexity of HEVC fractional interpolation is reduced. The PECR technique does not affect the PSNR and bit-rate. The PSCR technique slightly decreases PSNR and increases bit-rate.

In this paper, a low energy HEVC fractional (half-pixel and quarter-pixel) interpolation hardware for all PU sizes including the proposed techniques is also designed and implemented using Verilog HDL. The Verilog RTL code is verified to work at 125 MHz in a Xilinx Virtex 6 FPGA. The proposed hardware, in the worst case, can process 48 quad HD (2160x1600) video frames per second. The proposed PECR and PSCR techniques reduced the energy consumption of the proposed hardware up to 39.7% and 46.9%, respectively.

The rest of the paper is organized as follows. In Section II, HEVC fractional interpolation algorithm is explained. In Section III, the proposed PECR and PSCR techniques for HEVC fractional interpolation are explained. In Section IV, the proposed HEVC fractional interpolation hardware is explained and the implementation results are given. Section V presents the conclusion.

II. HEVC FRACTIONAL INTERPOLATION ALGORITHM

In HEVC, 3 different 8-taps FIR filters are used for both half-pixel and quarter-pixel interpolations. These 3 FIR filters type A, type B and type C are shown in (1), (2), and (3), respectively. The shift1 value is determined based on bit depth of the pixel. Integer pixels ($A_{x,y}$), half pixels ($a_{x,y}$, $b_{x,y}$, $c_{x,y}$, $d_{x,y}$, $h_{x,y}$, $n_{x,y}$) and quarter pixels ($e_{x,y}$, $f_{x,y}$, $g_{x,y}$, $i_{x,y}$, $j_{x,y}$, $k_{x,y}$, $p_{x,y}$, $q_{x,y}$, $r_{x,y}$) in a PU are shown in Fig. 1. The half-pixels a, b, c and d, h, n are interpolated from nearest integer pixels in the horizontal and vertical directions, respectively. The quarter-pixels e, f, g are interpolated from the nearest half pixels a, b, c respectively in the vertical direction using type A filter. The quarter-pixels i, j, k are interpolated similarly using type B filter. The quarter-pixels p, q, r are interpolated similarly using type C filter. HEVC fractional interpolation algorithm used in HEVC encoder calculates all fractional (half and quarter) pixels necessary for the fractional motion estimation process.

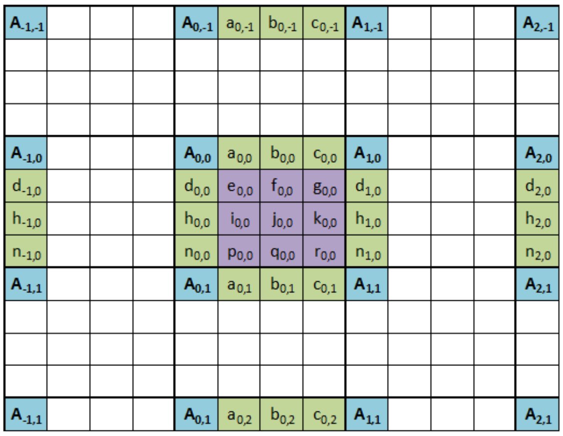
$A_{-1,-1}$ $A_{0,-1}$ $a_{0,-1}$ $b_{0,-1}$ $c_{0,-1}$ $A_{1,-1}$ $A_{2,-1}$
$A_{-1,0}$ $A_{0,0}$ $a_{0,0}$ $b_{0,0}$ $c_{0,0}$ $A_{1,0}$ $A_{2,0}$
$d_{-1,0}$ $d_{0,0}$ $e_{0,0}$ $f_{0,0}$ $g_{0,0}$ $d_{1,0}$ $d_{2,0}$
$h_{-1,0}$ $h_{0,0}$ $i_{0,0}$ $j_{0,0}$ $k_{0,0}$ $h_{1,0}$ $h_{2,0}$
$n_{-1,0}$ $n_{0,0}$ $p_{0,0}$ $q_{0,0}$ $r_{0,0}$ $n_{1,0}$ $n_{2,0}$
$A_{-1,1}$ $A_{0,1}$ $a_{0,1}$ $b_{0,1}$ $c_{0,1}$ $A_{1,1}$ $A_{2,1}$
$A_{-1,1}$ $A_{0,1}$ $a_{0,2}$ $b_{0,2}$ $c_{0,2}$ $A_{1,1}$ $A_{2,1}$

Fig. 1. Integer, Half and Quarter Pixels

$$a_{0,0}=\begin{pmatrix}-A_{-3,0}+4*A_{-2,0}-10*A_{-1,0}+58*A_{0,0}+\\ 17*A_{1,0}-5*A_{2,0}+A_{3,0}\end{pmatrix}\gg shift1 \quad (1)$$

$$b_{0,0}=\begin{pmatrix}-A_{-3,0}+4*A_{-2,0}-11*A_{-1,0}+40*A_{0,0}+40*A_{1,0}-\\ 11*A_{2,0}+4*A_{3,0}-A_{4,0}\end{pmatrix}\gg shift1 \quad (2)$$

$$c_{0,0}=\begin{pmatrix}A_{-2,0}-5*A_{-1,0}+17*A_{0,0}+58*A_{1,0}-\\ 10*A_{2,0}+4*A_{3,0}-A_{4,0}\end{pmatrix}\gg shift1 \quad (3)$$

III. Proposed PECR and PSCR Techniques

In this paper, two pixel correlation based computation and energy reduction techniques (PECR and PSCR) for HEVC fractional interpolation are proposed. The proposed PECR technique compares the input pixels of an FIR filter. If the input pixels are equal, the FIR filter output is equal to one of the input pixels. Therefore, the FIR filter calculation becomes unnecessary and it is skipped. If the input pixels are not equal, the FIR filter operation is performed.

The proposed PSCR technique compares the input pixels of an FIR filter. If the input pixels are similar, the FIR filter output is assumed to be equal to the input pixel multiplied with the largest coefficient in the FIR filter. Therefore, the FIR filter calculation becomes unnecessary and it is skipped. The PSCR technique checks the similarity of input pixels by truncating their least significant bits by specified amount (1, 2, 3 or 4 bits) and comparing the truncated pixels. If the input pixels are not similar, the FIR filter operation is performed.

Equality and similarity percentages of the input pixels of FIR filters vary from frame to frame. Therefore, one frame of Tennis, Kimono, Park Scene and BQ Terrace (1920x1080) videos [9] coded with quantization parameters (QP) 22, 27, 32 and 37 are analyzed to determine equality and similarity percentages using HEVC Test Model HM encoder software [10].

Table I shows the equality and 3-bit truncated similarity percentages for integer pixel inputs ($A_{x,y}$) and half-pixel inputs ($a_{x,y}$, $b_{x,y}$, $c_{x,y}$) of FIR filters. As shown in Table I, significant amount of FIR filter inputs are equal or similar. Therefore, the proposed PECR and PSCR techniques skip significant amount of FIR filter calculations.

Table II shows the addition and shift operation reductions achieved by the proposed PECR and PSCR for 3-bit truncated (3bT) techniques for one frame of each video sequence. As shown in Table II, the proposed PECR and PSCR for 3bT techniques achieved up to 26.34% and 49.28% computation reductions, respectively. The proposed techniques have overhead of only 3628800 comparisons for a full HD (1920x1080) frame.

The proposed PSCR technique is integrated into fractional interpolation performed by HEVC Test Model HM encoder software [10]. The impact of the proposed PSCR technique on rate-distortion performance is determined for Tennis, Kimono, Park Scene and BQ Terrace (1920x1080) videos [9]. Rate-distortion performances of original HEVC and HEVC using PSCR technique for fractional interpolation are shown in Fig. 2. The proposed PSCR technique slightly decreased PSNR and increased bit-rate.

TABLE I. Equality and Similarity Percentages

		HEVC Fractional Interpolation (Equal)				**HEVC Fractional Interpolation (3bT)**			
		A	**a**	**b**	**c**	**A**	**a**	**b**	**c**
Tennis	22	9.9	17.1	18.7	17.1	35.2	42.7	44.6	42.8
	27	13.8	24.8	25.5	24.7	37.4	45.4	47.4	45.5
	32	16.0	28.2	28.6	28.3	39.1	47.4	49.4	47.5
	37	18.9	31.3	31.2	31.4	40.5	50.0	52.1	50.1
Kimono	22	15.5	9.8	8.6	8.7	42.4	38.6	39.1	38.7
	27	17.2	11.1	10.3	10.1	45.7	41.5	42.1	41.5
	32	17.6	11.9	11.3	11.0	48.8	44.1	45.0	44.1
	37	19.5	12.6	12.0	11.7	52.3	46.9	47.9	47.0
Park Scene	22	4.8	2.4	2.0	2.3	30.8	28.8	30.0	28.8
	27	8.3	5.7	5.0	5.5	34.7	32.4	33.6	32.5
	32	10.2	7.7	6.8	7.5	37.9	35.5	36.9	35.6
	37	12.8	9.5	8.5	9.2	40.1	38.4	40.2	38.5
BQ Terrace	22	2.0	2.4	1.9	2.3	11.2	24.4	23.4	24.5
	27	7.3	6.0	5.3	5.9	21.2	34.2	32.8	34.3
	32	9.9	7.4	6.4	7.2	24.3	37.3	35.7	37.3
	37	11.9	9.5	8.4	9.3	26.6	39.3	37.4	39.4

TABLE II. Computation Reductions by PECR and PSCR 3bT

		PECR		**PSCR for 3bT**	
	QP	**Addition Reduction**	**Shift Reduction**	**Addition Reduction**	**Shift Reduction**
Tennis	**22**	14.54 %	14.54 %	40.10 %	40.10 %
	37	26.34 %	26.34 %	46.64 %	46.64 %
Kimono	**22**	11.62 %	11.62 %	40.24 %	40.24 %
	37	15.06 %	15.06 %	49.28 %	49.28 %
Park Scene	**22**	3.26 %	3.26 %	29.84 %	29.84 %
	37	10.56 %	10.56 %	39.46 %	39.46 %
BQ Terrace	**22**	2.12 %	2.12 %	18.94 %	18.94 %
	37	10.20 %	10.20 %	33.86 %	33.86 %

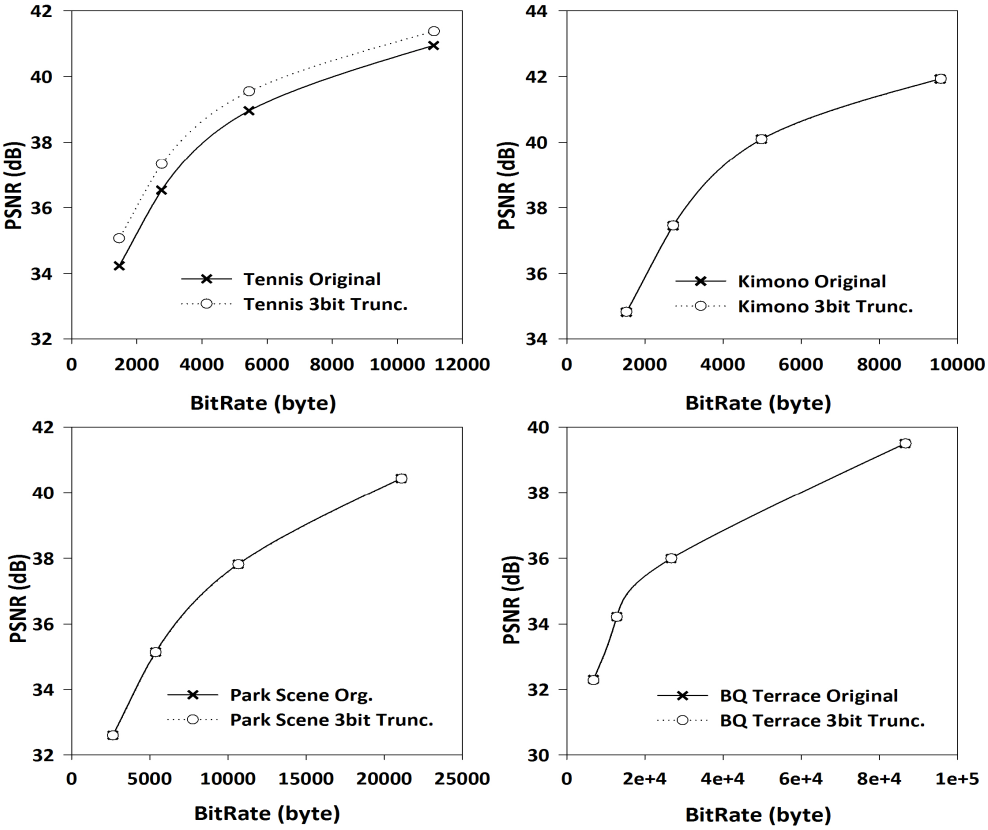

Fig. 2. Rate-Distortion Performances of Original HEVC and HEVC using PSCR Technique for Fractional Interpolation

IV. Proposed HEVC Fractional Interpolation Hardware

The proposed HEVC fractional interpolation hardware for all PU sizes including the proposed PECR and PSCR techniques is shown in Fig. 3. The proposed hardware interpolates all the fractional (half-pixels and quarter-pixels) pixels for the luma component of a PU using integer or half pixels. Four buffers are used to store integer and half pixels necessary for interpolating the half and quarter pixels. The interpolated a, b and c half-pixels are stored in the filtered pixels buffers A, B and C, respectively. These on-chip buffers reduce the required off-chip memory bandwidth and power consumption.

8 parallel interpolation units are used to interpolate the 8x3=24 fractional pixels of a PU in parallel. As shown in Fig. 3, three FIR filters (type A, type B, type C) are implemented separately in an interpolation unit.

Since 15 fractional pixels should be interpolated for one integer pixel, 64x15 fractional pixels should be interpolated for an 8x8 PU. Also, 8x7 extra a, b, c half-pixels should be interpolated for the interpolation of quarter-pixels. First, integer pixels are loaded into integer pixel buffer in one clock cycle. Then, 8x8 d, h, n half-pixels are interpolated and stored in the output buffer in 8 clock cycles. After that 15x8 a, b, c half-pixels are interpolated and stored in the filtered pixel buffers A, B and C, respectively, in 15 clock cycles. Finally, 9x8x8 quarter-pixels are interpolated using a, b, c half-pixels and stored in the output buffer in 3x8=24 clock cycles. Therefore, the proposed hardware, in the worst case, interpolates the fractional pixels for an 8x8 PU in 48 clock cycles.

The original HEVC fractional interpolation hardware (FIHW) does not have the comparison unit. In both the proposed HEVC fractional interpolation hardware including the PECR technique (FIHW+PECR) and the proposed HEVC fractional interpolation hardware including the PSCR technique (FIHW+PSCR), 14 comparators are used to check similarity of the input pixels of FIR filters. FIHW+PECR uses 8-bit comparators. FIHW+PSCR for 1bT uses 7-bit comparators. Similarly, FIHW+PSCR for 4bT uses 4-bit comparators. Based on the comparison results, disable signals are generated for each FIR filter and sent to the interpolation units. If the input pixels of an FIR filter are equal or similar, input registers of the corresponding FIR filter hardware are not updated, and a multiplexer at the output of interpolation unit is used to select the input pixel multiplied with the largest coefficient in the FIR filter instead of interpolated pixel. This prevents unnecessary switching activities in the FIR filter hardware.

The proposed FIHW, FIHW+PECR and FIHW+PSCR hardware are implemented using Verilog HDL. The Verilog RTL codes are verified with RTL simulations. RTL simulation results matched the results of fractional interpolation implementation in HEVC HM encoder software [10].

The Verilog RTL codes are mapped to a Xilinx XC6VLX75T FF784 FPGA with speed grade 3 using Xilinx ISE 13.4. All FPGA implementations are verified to work at 125 MHz by post place and route simulations. Post place and

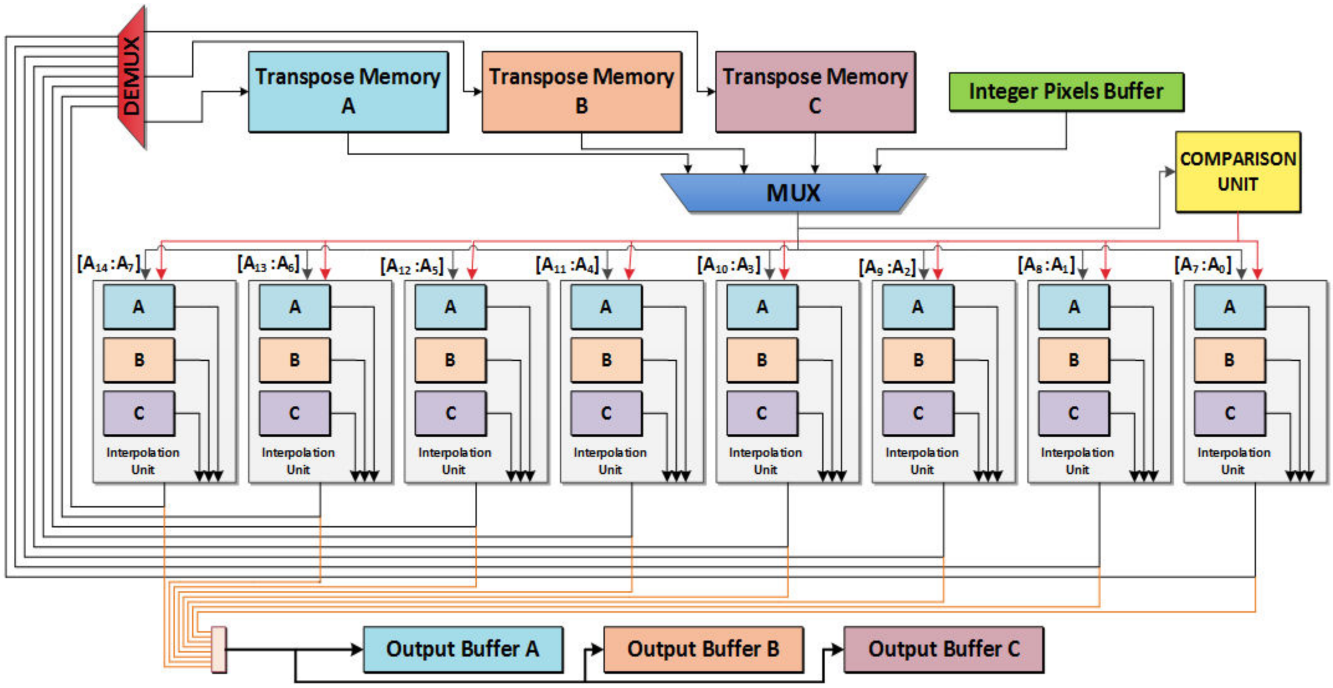

Fig. 3. Proposed HEVC Fractional Interpolation Hardware

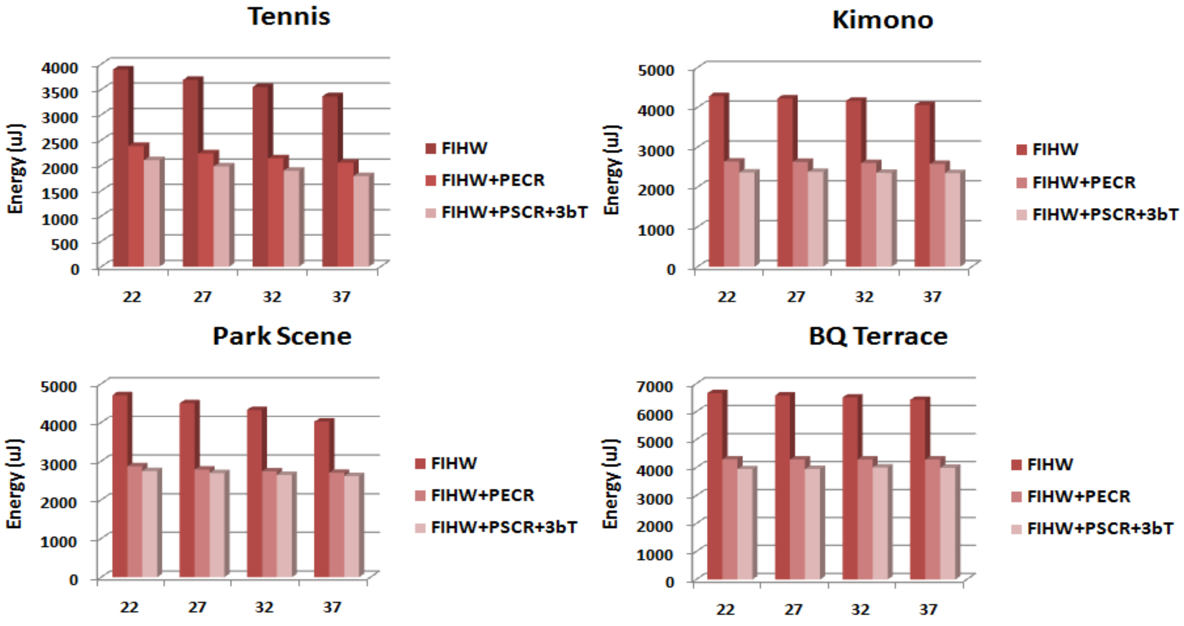

Fig. 4. Energy Consumptions of HEVC Fractional Interpolation Hardware

route simulation results matched the results of fractional interpolation implementation in HEVC HM encoder software [10]. Therefore, they can process 48 quad HD (2160x1600) video frames per second. FIHW FPGA implementation uses 4110 LUTs, 3448 DFFs and 6 BRAMs. FIHW+PECR FPGA implementation uses 4577 LUTs, 3408 DFFs, and 4 BRAMs. FIHW+PSCR for 3bT FPGA implementation uses 2381 LUTs, 849 DFFs, and 4 BRAMs.

Power consumptions of FIHW, FIHW+PECR and FIHW+PSCR for 3bT FPGA implementations are estimated using Xilinx XPower Analyzer tool. Post place and route timing simulations are performed for Tennis, Kimono, Park Scene and BQ Terrace (1920x1080) videos at 100 MHz [9], and signal activities are stored in VCD files. These VCD files are used for estimating the power consumptions of all FPGA implementations. Energy consumption results of FIHW, FIHW+PECR and FIHW+PSCR for 3bT for one frame of each video are shown in Fig. 4. As shown in Fig. 4, PECR and PSCR techniques reduced the energy consumption of FIHW FPGA implementation up to 39.7% and 46.9%, respectively.

V. Conclusion

In this paper, two pixel correlation based computation and energy reduction techniques, PECR and PSCR, for HEVC fractional interpolation are proposed. In this paper, a low energy HEVC fractional interpolation hardware for all PU sizes including the proposed techniques is also designed and implemented using Verilog HDL. The proposed hardware, in the worst case, can process 48 quad HD (2160x1600) video frames per second. The proposed PECR and PSCR techniques reduced the energy consumption of this hardware up to 39.7% and 46.9%, respectively.

Acknowledgement

This research was supported in part by the Scientific and Technological Research Council of Turkey (TUBITAK) under the contract 115E290.

References

[1] High Efficiency Video Coding, ITU-T Rec. H.265 and ISO/IEC 23008-2 (HEVC), ITU-T and ISO/IEC, Apr. 2013.

[2] E. Kalali, A. C. Mert, I. Hamzaoglu, "A computation and energy reduction technique for HEVC Discrete Cosine Transform", IEEE Trans. on Consumer Electronics, vol. 62, no. 2, pp. 166-174, May 2016.

[3] A. C. Mert, E. Kalali, I. Hamzaoglu, "Low Complexity HEVC Sub-Pixel Motion Estimation Technique and Its Hardware Implementation", IEEE Int. Conference on Consumer Electronics – Berlin, Sept. 2016.

[4] E. Kalali, Y. Adibelli, I. Hamzaoglu, "A High Performance and Low Energy Intra Prediction Hardware for High Efficiency Video Coding", Int. Conference on Field Programmable Logic and Applications (FPL), Aug. 2012.

[5] E. Kalali, Y. Adibelli, I. Hamzaoglu, "A Reconfigurable HEVC Sub-Pixel Interpolation Hardware", IEEE Int. Conference on Consumer Electronics - Berlin, Sep. 2013.

[6] E. Kalali, E. Ozcan, O. M. Yalcinkaya, I Hamzaoglu, "A Low Energy HEVC Inverse Transform Hardware", IEEE Trans. on Consumer Electronics, vol. 60, no. 4, pp. 754-761, Nov. 2014.

[7] J. Vanne, M. Viitanen, T. D. Hamalainen, A. Hallapuro, "Comparative Rate-Distortion-Complexity Analysis of HEVC and AVC Video Codecs", IEEE Trans. on CAS for Video Technology, vol. 22, no. 12, pp. 1885-1898, Dec. 2012.

[8] S. Yalcin, I. Hamzaoglu, "A High Performance Hardware Architecture for Half-Pixel Accurate H.264 Motion Estimation", IFIP Int. Conference on VLSI-SoC, Oct. 2006.

[9] F. Bossen, "Common test conditions and software reference configurations", JCTVC-I1100, May 2012.

[10] K. McCann, B. Bross, W.J. Han, I.K. Kim, K. Sugimoto, G. J. Sullivan, "High Efficiency Video Coding (HEVC) Test Model (HM) 15 Encoder Description", JCTVC-Q1002, June 2014.