Approximate HEVC Fractional Interpolation Filters and Their Hardware Implementations

Ercan Kalali and Ilker Hamzaoglu, *Senior Member, IEEE*

***Abstract*—High efficiency video coding (HEVC) fractional interpolation algorithm has very high computational complexity. In this paper, two approximate HEVC fractional interpolation filters are proposed. They significantly reduce computational complexity of HEVC fractional interpolation with a negligible PSNR loss and bit rate increase. In addition, two approximate HEVC fractional interpolation hardware are proposed. They, in the worst case, can process 45 quad full HD (3840 x 2160) fps. They consume up to 67.1% less energy than original HEVC fractional interpolation hardware. Therefore, they can be used in consumer electronics products that require a low energy HEVC encoder.**

***Index Terms*—HEVC, fractional interpolation, hardware implementation, low energy.**

I. Introduction

HIGH efficiency video coding (HEVC) achieves 50% more compression than H.264. But, it has higher computational complexity than H.264 [1]–[7]. HEVC fractional interpolation algorithm has very high computational complexity [8]. 25% of HEVC encoder complexity and half of HEVC decoder complexity are caused by fractional interpolation.

H.264 uses a 6-tap FIR filter to interpolate half pixels. It uses linear interpolation to interpolate quarter pixels. HEVC uses same interpolation filters to interpolate both half and quarter pixels [9]. It uses two different 7-tap and one 8-tap FIR filters to interpolate fractional pixels. H.264 uses 4×4 and 16×16 block sizes for fractional interpolation. HEVC uses 4×4 to 64×64 prediction unit (PU) sizes for fractional interpolation. Therefore, HEVC fractional interpolation has higher computational complexity than H.264 fractional interpolation.

In this paper, two approximate HEVC fractional interpolation filters (F1 and F2) are proposed. Both F1 and F2 use one 4-tap and two different 3-tap FIR filters instead of using one 8-tap and two different 7-tap FIR filters. The proposed interpolation filters significantly reduce computational complexity of HEVC fractional interpolation with a negligible PSNR loss and bit rate increase. F2 reduces computational complexity more than F1 with more PSNR loss and bit rate increase.

The proposed approximate fractional interpolation filters are used in fractional motion estimation stage of an HEVC encoder. After best fractional motion vector is determined, original HEVC fractional interpolation filter is used in coding stage of the HEVC encoder. Therefore, the proposed approximate fractional interpolation filters do not cause encoder-decoder mismatch.

In this paper, two approximate HEVC fractional interpolation hardware for all PU sizes are designed and implemented using Verilog HDL for each proposed approximate fractional interpolation filter. The first hardware implements multiplications with constant coefficients using adders and shifters. The second hardware implements addition and shift operations using Hcub multiplierless constant multiplication (MCM) algorithm [10]. The second hardware for both F1 and F2, in the worst case, can process 45 quad full HD (QFHD) frames per second (fps). They consume up to 67.1% less energy than original HEVC fractional interpolation hardware. F2 fractional interpolation hardware has smaller area and lower energy consumption than F1 fractional interpolation hardware.

Approximate HEVC fractional interpolation filters are proposed in [11] and [12]. However, the approximate HEVC fractional interpolation filters proposed in this paper have less computational complexity and better rate-distortion performance than the ones proposed in [11] and [12].

Several HEVC fractional interpolation hardware are proposed in [10] and [13]–[20]. The approximate HEVC fractional interpolation hardware proposed in this paper are similar to the HEVC fractional interpolation hardware proposed in [10]. In Section IV, they are compared with the HEVC fractional interpolation hardware proposed in [10] and [13]–[20]. They have much smaller hardware area and lower power consumption than the other hardware. Only the HEVC fractional interpolation hardware proposed in [15], [17], and [18] have higher throughput than them at the expense of much larger hardware area.

The rest of this paper is organized as follows. Section II explains the HEVC fractional interpolation algorithm. The proposed approximate HEVC fractional interpolation filters are explained in Section III. Section IV presents the proposed approximate HEVC fractional interpolation hardware and reports their implementation results. Section V presents conclusions.

Manuscript received March 31, 2018; revised May 27, 2018; accepted August 18, 2018. Date of publication August 29, 2018; date of current version September 24, 2018. This work was supported by the Scientific and Technological Research Council of Turkey under Contract 115E290. *(Corresponding author: Ercan Kalali.)*

The authors are with the Faculty of Engineering and Natural Sciences, Sabanci University, 34956 Istanbul, Turkey (e-mail: ercankalali@sabanciuniv.edu; hamzaoglu@sabanciuniv.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCE.2018.2867806

$$a_{0,0} = \left(-A_{-3,0} + 4 * A_{-2,0} - 10 * A_{-1,0} + 58 * A_{0,0} + 17 * A_{1,0} - 5 * A_{2,0} + A_{3,0}\right) \gg 6 \quad (1)$$

$$b_{0,0} = \left(-A_{-3,0} + 4 * A_{-2,0} - 11 * A_{-1,0} + 40 * A_{0,0} + 40 * A_{1,0} - 11 * A_{2,0} + 4 * A_{3,0} - A_{4,0}\right) \gg 6 \quad (2)$$

$$c_{0,0} = \left(A_{-2,0} - 5 * A_{-1,0} + 17 * A_{0,0} + 58 * A_{1,0} - 10 * A_{2,0} + 4 * A_{3,0} - A_{4,0}\right) \gg 6 \quad (3)$$

$$a_{0,0} = \left(-7 * A_{-1,0} + 58 * A_{0,0} + 13 * A_{1,0}\right) \gg 6 \quad (4)$$

$$b_{0,0} = \left(-8 * A_{-1,0} + 40 * A_{0,0} + 40 * A_{1,0} - 8 * A_{2,0}\right) \gg 6 \quad (5)$$

$$c_{0,0} = \left(13 * A_{-0,0} + 58 * A_{1,0} - 7 * A_{2,0}\right) \gg 6 \quad (6)$$

$$a_{0,0} = \left(-8 * A_{-1,0} + 64 * A_{0,0} + 8 * A_{1,0}\right) \gg 6 \quad (7)$$

$$b_{0,0} = \left(-8 * A_{-1,0} + 40 * A_{0,0} + 40 * A_{1,0} - 8 * A_{2,0}\right) \gg 6 \quad (8)$$

$$c_{0,0} = \left(8 * A_{0,0} + 64 * A_{1,0} - 8 * A_{2,0}\right) \gg 6. \quad (9)$$

II. HEVC Fractional Interpolation Algorithm

In HEVC standard, both half pixels and quarter pixels are interpolated using one 8-tap and two different 7-tap FIR filters. These three FIR filters (A, B, C) are shown in (1)-(3), respectively. The symbol ($\gg$) in the equations represents right shift operation which is used to reduce bit length of output fractional pixel to 8 bits.

Integer pixels ($A_{x,y}$) and fractional pixels in a PU are shown in Fig. 1. Half pixels a, b, c are interpolated from horizontal integer pixels using FIR filters A, B, C respectively. Half pixels d, h, n are interpolated from vertical integer pixels using FIR filters A, B, C respectively. Quarter pixels e, f, g are interpolated from half pixels a, b, c respectively using FIR filter A. Quarter pixels i, j, k and p, q, r are interpolated similarly using FIR filters B and C, respectively.

All the fractional pixels necessary for fractional motion estimation are calculated in HEVC fractional interpolation algorithm used in HEVC encoder.

III. Proposed Approximate HEVC Fractional Interpolation Filters

In this paper, two approximate HEVC fractional interpolation filters (F1 and F2) are proposed. Both F1 and F2 use one 4-tap and two different 3-tap FIR filters. But, they use different filter coefficients. The proposed approximate HEVC fractional interpolation filter equations for F1 and F2 are shown in (4)-(6) and (7)-(9), respectively.

In original HEVC FIR filter A, if values of the pixels ($A_{-3,0}$, $A_{-2,0}$, $A_{-1,0}$) multiplied with first three coefficients (-1, 4, -10) are the same, multiplication and addition result can be calculated by multiplying one pixel with -7 ($-1 + 4 - 10 = -7$). In original HEVC fractional interpolation filters, small coefficients have less effect on the filter result. In addition, since the pixels multiplied with small coefficients are neighboring pixels, because of spatial correlation, their values will be very similar. Therefore, the coefficients of F1 are determined by assuming that values of the pixels multiplied with small coefficients are the same. The coefficients of F2 are determined by replacing the coefficients of F1 with closest 2^n values. In this way, multiplications with coefficients of F2 are performed using shift operations. In addition, F1 and F2 have similar frequency responses with original HEVC fractional interpolation filters.

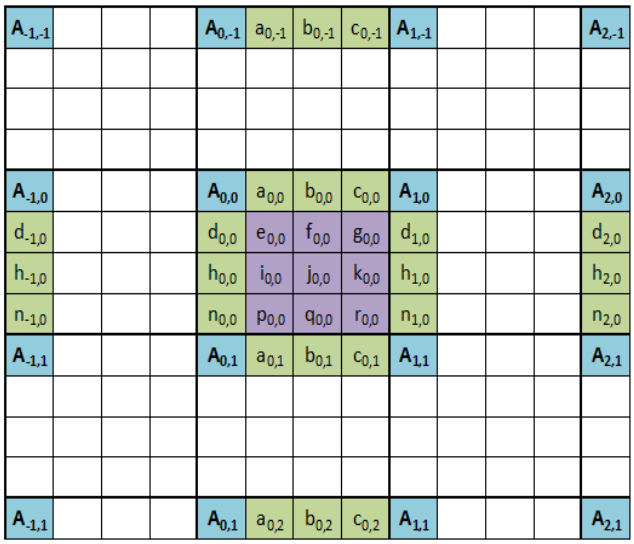

Fig. 1. Integer and fractional pixels.

TABLE I
Addition and Shift Reductions

Filter		A		B		C		Avg. (%)
		Add	Shift	Add	Shift	Add	Shift	
Original		11	8	13	10	11	8	
F1	Num.	7	6	5	6	7	6	
	Red. (%)	36.3	25.0	61.5	40.0	36.3	25.0	37.4
F2	Num.	2	3	5	6	2	3	
	Red. (%)	81.8	62.5	61.5	40.0	81.8	62.5	65.0
[11]	Num.	11	6	11	8	11	6	
	Red. (%)	0.0	25.0	15.4	20.0	0.0	25.0	14.2
[12]	Num.	9	6	9	10	9	6	
	Red. (%)	18.2	25.0	30.8	0.0	18.2	25.0	19.5

Table I shows the number of addition and shift operations necessary for calculating FIR filters used in HEVC fractional interpolation (Orig.), FIR filters used in the proposed approximate HEVC fractional interpolation (F1 and F2), and FIR filters used in the approximate HEVC fractional interpolation proposed in [11] and [12].

The proposed approximate HEVC fractional interpolation filters (F1 and F2) are integrated into fractional motion estimation in HEVC HM software encoder 15.0 [21]. First ten frames of some of the HEVC test videos [22] are coded with low delay P (LP) test configuration and with four different quantization parameters (QP) using HEVC HM 15.0 with original

TABLE II
BD-RATE (%) AND BD-PSNR (DB) RESULTS

Video Sequence		F1		F2		[11]		[12]	
		BD-Rate	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR
2560x1600	People on Street	-0.27	0.01	1.13	-0.05	---	---	---	---
	Traffic	0.51	-0.02	1.56	-0.06	---	---	---	---
1920x1080	Tennis	-0.01	0.01	0.76	-0.02	---	---	---	---
	Kimono	-0.31	0.01	0.31	-0.01	1.79	-0.06	1.05	-0.03
	Basketball Drive	0.76	-0.01	1.46	-0.03	1.22	-0.03	1.41	-0.03
	Park Scene	0.73	-0.03	1.77	-0.06	2.42	-0.08	3.77	-0.11
1280x720	Vidyo1	0.17	-0.01	0.60	-0.02	---	---	---	---
	Vidyo4	0.25	-0.01	0.49	-0.01	---	---	---	---
	Kristen and Sara	0.53	-0.02	1.14	-0.04	3.87	-0.12	4.12	-0.12
	Four People	0.08	0.00	0.48	-0.02	3.25	-0.11	3.02	-0.10
832x480	Keiba	0.16	-0.01	1.36	-0.05	---	---	---	---
	BQ Mall	0.79	-0.04	1.36	-0.06	1.69	-0.07	3.73	-0.14
	Race Horses	0.61	-0.03	1.91	-0.09	1.28	-0.05	2.21	-0.08
	Basketball Drill	1.56	-0.06	1.64	-0.07	0.35	-0.01	1.28	-0.05
Average		0.40	-0.01	1.14	-0.04	1.98	-0.07	2.57	-0.08

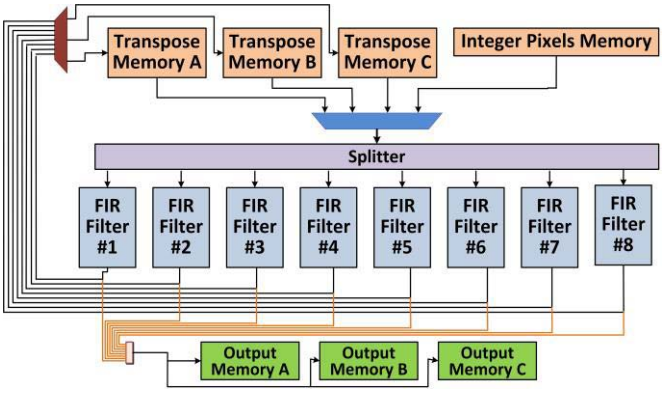

Fig. 2. Proposed AS approximate HEVC fractional interpolation hardware.

HEVC fractional interpolation filters, F1 and F2. The resulting rate-distortion performances are shown in Table II.

The proposed F1 and F2 filters significantly reduce computational complexity of HEVC fractional interpolation with a negligible PSNR loss and bit rate increase. They have less computational complexity and better rate-distortion performance than the ones proposed in [11] and [12].

IV. PROPOSED APPROXIMATE HEVC FRACTIONAL INTERPOLATION HARDWARE

In this paper, two approximate HEVC fractional interpolation hardware for all PU sizes are designed for each proposed approximate interpolation filter. The first hardware (AS) implements multiplications with constant coefficients using adders and shifters. In this hardware, three different datapaths are used for implementing A, B and C FIR filters. It interpolates $8 \times 3 = 24$ fractional pixels in parallel using 24 (8 A, 8 B, 8 C) parallel datapaths. The proposed AS approximate HEVC fractional interpolation hardware is shown in Fig. 2.

Since different fractional interpolation filter equations multiply same integer pixel with different constant coefficients, in the second hardware (MCM), Hcub MCM algorithm is used for reducing number and size of the adders [10]. A multiplier block (MB) hardware is given one input. It outputs multiplications of this input with all the constants. The proposed MCM approximate HEVC fractional interpolation hardware is shown in Fig. 3.

Integer pixels are stored in one on-chip memory. Then, half pixels (a, b, c) that will be used for interpolating the quarter pixels are stored in three on chip memories. Since a, b, c half pixels are interpolated in horizontal direction and used in vertical direction for quarter pixel interpolations, transpose memory architecture is used to store a, b, c half pixels.

Both proposed MCM hardware implementing the proposed F1 fractional interpolation filter (F1 MCM hardware) and proposed MCM hardware implementing the proposed F2 fractional interpolation filter (F2 MCM hardware) interpolate $8 \times 3 = 24$ fractional pixels in parallel. First, multiplier blocks perform multiplications with constant coefficients. Then, fractional pixels are calculated using adder trees. Since different constant coefficients are used in F1 and F2 filters, different multiplier blocks are used in F1 MCM hardware and F2 MCM hardware.

Since the proposed approximate HEVC fractional interpolation filters F1 and F2 use FIR filters with less number of taps than the original HEVC fractional interpolation filter, the proposed F1 AS, F1 MCM, F2 AS and F2 MCM hardware need to access 11 pixels instead of 15 pixels in order to interpolate $8 \times 3 = 24$ fractional pixels. Therefore, they require less memory accesses than the original HEVC fractional interpolation hardware.

F1 AS, F2 AS, F1 MCM and F2 MCM hardware interpolate the fractional pixels for an 8×8 PU in 44 clock cycles. First, 8×8 half pixels are interpolated. Then, 8×11 half pixels that will be used for interpolating the quarter pixels are interpolated. Finally, 64×9 quarter pixels are interpolated. Scheduling of memory read and interpolation operations in F1 AS, F2 AS, F1 MCM and F2 MCM hardware are shown in Fig. 4.

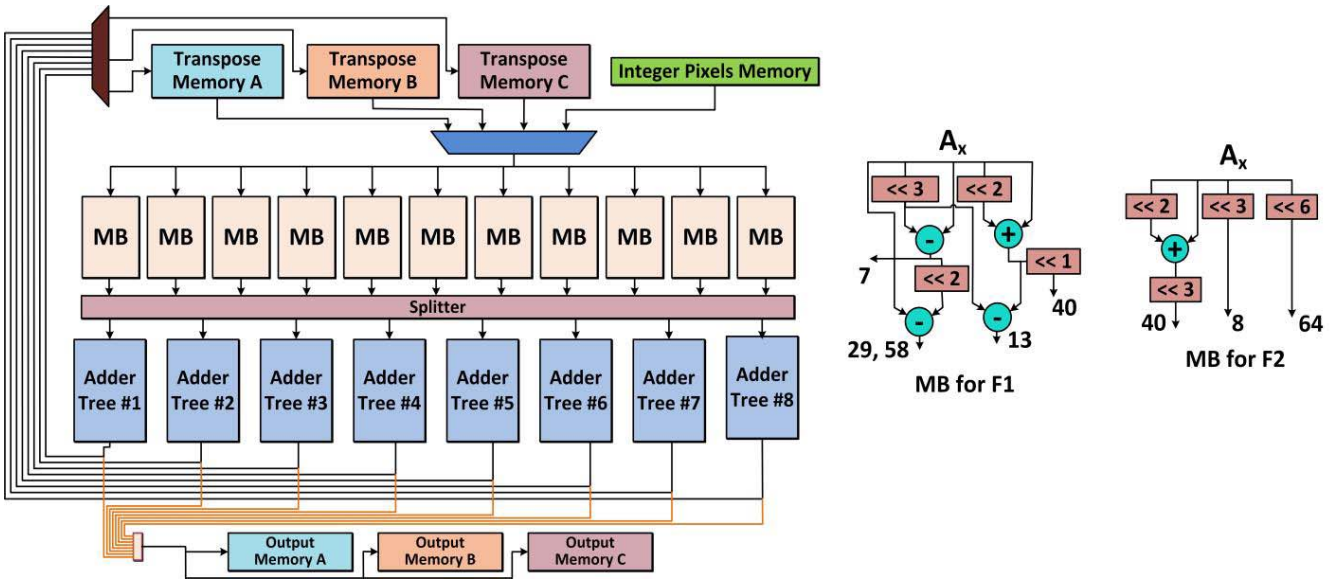

Fig. 3. Proposed MCM approximate HEVC fractional interpolation hardware.

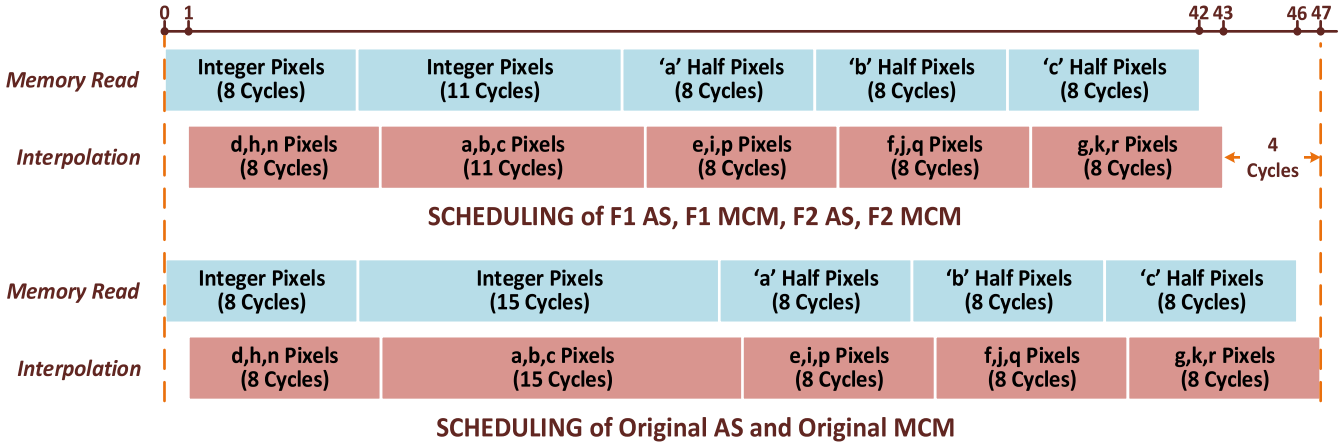

Fig. 4. Scheduling of HEVC fractional interpolation hardware.

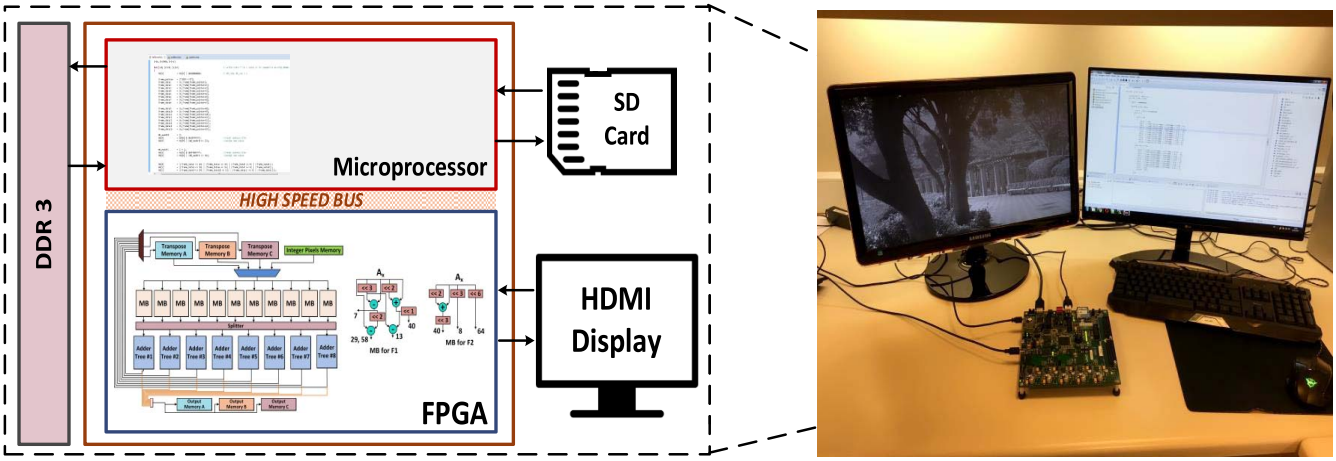

Fig. 5. Implementation of proposed approximate HEVC fractional interpolation hardware on an FPGA board.

The proposed F1 AS, F1 MCM, F2 AS and F2 MCM hardware are implemented using Verilog HDL. The Verilog RTL codes are synthesized, placed and routed to a 40 nm FPGA. FPGA implementations are verified with both RTL and post place & route timing simulations. The simulation results matched the results of HEVC HM software encoder [21].

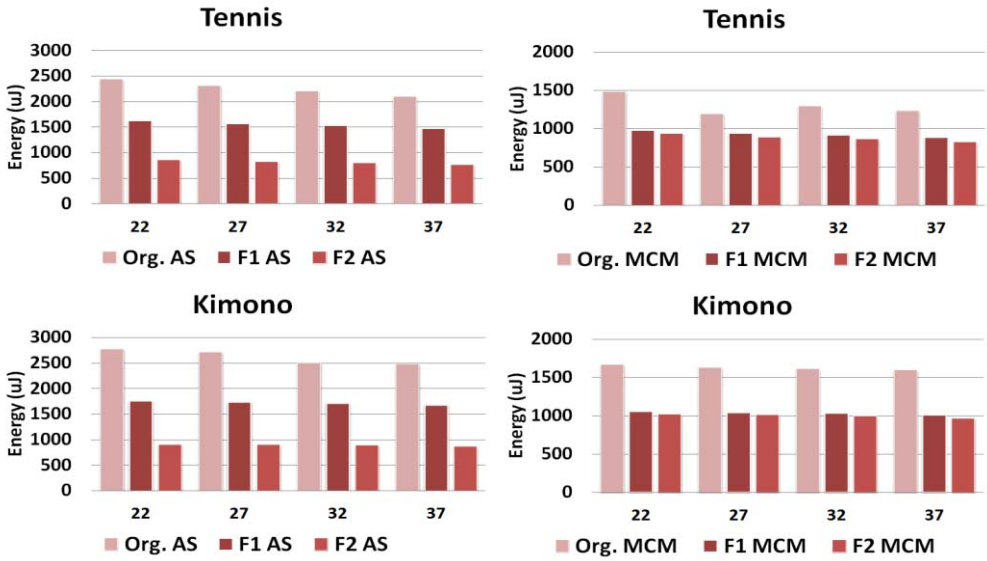

Fig. 6. Energy consumption results.

TABLE III
FPGA IMPLEMENTATION RESULTS

	Original [10]		Proposed F1		Proposed F2	
	AS	MCM	AS	MCM	AS	MCM
Slice	1669	1557	1144	834	963	731
LUT	4110	3929	2416	2008	1601	1567
DFF	3448	3422	2596	3034	1873	2762
BRAM	6	6	6	6	6	6
Freq. (MHz)	200	200	200	278	250	278
Fps	30 Quad Full HD	30 Quad Full HD	33 Quad Full HD	45 Quad Full HD	41 Quad Full HD	45 Quad Full HD
Power Dissip.	152 mW	93 mW	104 mW	88 mW	67 mW	80 mW

TABLE IV
ASIC IMPLEMENTATION RESULTS

	Original [10]		Proposed F1		Proposed F2	
	AS	MCM	AS	MCM	AS	MCM
Tech.	90 nm	90 nm	90 nm	90 nm	90 nm	90 nm
Gate Count	29.5 K	28.5 K	13.2 K	12.8 K	10.6 K	11.2 K
Freq. (MHz)	250	250	300	300	300	300
Fps	37 Quad Full HD	37 Quad Full HD	49 Quad Full HD	49 Quad Full HD	49 Quad Full HD	49 Quad Full HD
Power Dissip.	27.3 mW	23.9 mW	16.4 mW	15.8 mW	14.8 mW	14.9 mW

FPGA implementations are also verified on an FPGA board as shown in Fig. 5. The FPGA board has a 28 nm FPGA and dual-core microprocessor. It also has 1GB DRAM and several interfaces such as UART and HDMI. Microprocessor reads video frames from SD card and sends them to FPGA using a high speed bus. The proposed hardware interpolates the video frames. Then, microprocessor displays interpolated frames on HDMI monitor and stores them to SD card.

FPGA implementation results are shown in Table III. F1 AS implementation can work at 200 MHz, and it can process 33 QFHD (3840×2160) fps. F2 AS implementation can work at 250 MHz, and it can process 41 QFHD fps. F1 MCM and F2 MCM implementations can work at 278 MHz, and they can process 45 QFHD fps. The proposed F1 and F2 approximate HEVC fractional interpolation hardware are faster and smaller than the original HEVC fractional interpolation hardware proposed in [10].

The Verilog RTL codes of the proposed F1 AS, F1 MCM, F2 AS and F2 MCM hardware are synthesized, placed and routed to a 90nm standard cell library as well. The gate counts of these ASIC implementations are calculated based on 2 $\times$ 1 NAND gate area. ASIC implementation results are shown in Table IV.

Power consumptions of F1 AS, F2 AS, F1 MCM and F2 MCM are estimated for Tennis and Kimono (1920×1080) videos [22] using a gate level power estimation tool. Signal activities captured during post place & route timing simulations are used to estimate power consumptions. Energy consumptions of all FPGA implementations are shown in Fig. 6. The proposed approximate HEVC fractional interpolation hardware consume up to 67.1% less energy than the original HEVC fractional interpolation hardware proposed in [10].

The proposed approximate HEVC fractional interpolation hardware are compared with the HEVC fractional interpolation hardware proposed in [10] and [13]–[20]. The comparisons of ASIC and FPGA implementations are shown in Table V and Table VI, respectively. Some of the results are not given in Table V and Table VI, because they are not available in [13]–[19].

A coarse grained reconfigurable datapath is proposed to reduce area and adaptive scheduling is proposed to increase

TABLE V
COMPARISON OF ASIC IMPLEMENTATIONS

	[10]	[13]	[14]	[15]	[16]	[17]	[18]	F1	F2
Technology	90 nm	150 nm	90 nm	90 nm	90 nm	130 nm	40 nm	90 nm	90 nm
Gate Count	28.5 K	30.2 K	224 K	383 K	37.2 K	126.8 K	297.3 K	12.8 K	11.2 K
Max. Freq. (MHz)	200	312	333	192	240	208	342	300	300
Fps	30 3840x2160	30 3840x2160	30 1920x1080	60 3840x2160	47 3840x2160	86 3840x2160	60 7680x4320	49 3840x2160	49 3840x2160
Power Dissipation	23.9 mW	---	---	---	---	---	48.1 mW	15.8 mW	14.9 mW

TABLE VI
COMPARISON OF FPGA IMPLEMENTATIONS

	[10]	[15]	[19]	[20]	F1	F2
FPGA	40 nm FPGA	65 nm FPGA	40 nm FPGA	65 nm FPGA	40 nm FPGA	40 nm FPGA
Slice Count	1557	---	---	2181	834	731
LUT Count	3929	28486	24202	5017	2008	1567
Max. Freq. (MHz)	200	120	200	283	278	278
Fps	30 3840x2160	---	60 1920x1080	30 2560x1600	45 3840x2160	45 3840x2160
Power Dissipation	93 mW	---	171 mW	89 mW	88 mW	80 mW

throughput in [13]. A fractional interpolation hardware is proposed for HEVC encoder in [14]. Data-reuse technique is used to reduce memory accesses and highly-parallel architecture is used to increase throughput in [15]. Efficient memory organization and reuse of datapath are proposed in [16]. Resource sharing for common partial terms of the interpolation filters is proposed in [17]. A fractional interpolation hardware is proposed for motion compensation in [18]. Many parallel interpolation hardware are used in [19]. Reconfigurable interpolation datapaths are used to reduce area and power consumption in [20].

The proposed approximate HEVC fractional interpolation hardware have much smaller hardware area and lower power consumption than the other hardware. The smallest hardware in the literature has more than two times larger area than the proposed hardware. Only the HEVC fractional interpolation hardware proposed in [15], [17], and [18] have higher throughput than them. However, they have more than ten times larger area than the proposed hardware. In addition, performance result of the hardware proposed in [18] is given for motion compensation. Performance results of the rest of the hardware including the ones proposed in this paper are given for motion estimation.

V. CONCLUSION

In this paper, two approximate HEVC fractional interpolation filters (F1 and F2) are proposed. They significantly reduce computational complexity of HEVC fractional interpolation by modifying the interpolation filter coefficients at the expense of a negligible PSNR loss and bit rate increase. F2 filter reduces computational complexity more than F1 filter with more PSNR loss and bit rate increase. In addition, two approximate HEVC fractional interpolation hardware are proposed. They have the smallest hardware area in the literature. F2 hardware has smaller area than F1 hardware. They, in the worst case, can process 45 QFHD (3840×2160) fps. They consume up to 67.1% less energy than original HEVC fractional interpolation hardware.

REFERENCES

[1] "High efficiency video coding," Int. Telecommun. Union, Geneva, Switzerland, ITU-Recommendation H.265, Apr. 2013.

[2] H. Azgin, E. Kalali, and I. Hamzaoglu, "A computation and energy reduction technique for HEVC intra prediction," *IEEE Trans. Consum. Electron.*, vol. 63, no. 1, pp. 36–43, Feb. 2017.

[3] E. Kalali, A. C. Mert, and I. Hamzaoglu, "A computation and energy reduction technique for HEVC discrete cosine transform," *IEEE Trans. Consum. Electron.*, vol. 62, no. 2, pp. 166–174, May 2016.

[4] E. Kalali, E. Ozcan, O. M. Yalcinkaya, and I. Hamzaoglu, "A low energy HEVC inverse transform hardware," *IEEE Trans. Consum. Electron.*, vol. 60, no. 4, pp. 754–761, Nov. 2014.

[5] E. Ozcan, E. Kalali, Y. Adibelli, and I. Hamzaoglu, "A computation and energy reduction technique for HEVC intra mode decision," *IEEE Trans. Consum. Electron.*, vol. 60, no. 4, pp. 745–753, Nov. 2014.

[6] E. Ozcan, Y. Adibelli, and I. Hamzaoglu, "A high performance deblocking filter hardware for high efficiency video coding," *IEEE Trans. Consum. Electron.*, vol. 59, no. 3, pp. 714–720, Aug. 2013.

[7] F. Pescador, M. Chavarrias, M. J. Garrido, E. Juarez, and C. Sanz, "Complexity analysis of an HEVC decoder based on a digital signal processor," *IEEE Trans. Consum. Electron.*, vol. 59, no. 2, pp. 391–399, May 2013.

[8] J. Vanne, M. Viitanen, T. D. Hamalainen, and A. Hallapuro, "Comparative rate-distortion-complexity analysis of HEVC and AVC video codecs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1885–1898, Dec. 2012.

[9] E. Kalali, Y. Adibelli, and I. Hamzaoglu, "A reconfigurable HEVC sub-pixel interpolation hardware," in *Proc. IEEE Int. Conf. Consum. Electron. Berlin*, Berlin, Germany, 2013, pp. 125–128.

[10] E. Kalali and I. Hamzaoglu, "A low energy HEVC sub-pixel interpolation hardware," in *Proc. IEEE Int. Conf. Image Process.*, Paris, France, 2014, pp. 1218–1222.

[11] A. Diefy, A. Shalaby, and M. S. Sayed, "Efficient architectures for HEVC luma interpolation filter," in *Proc. Int. Conf. Microelectron.*, Casablanca, Morocco, 2015, pp. 9–12.

[12] A. Diefy, A. Shalaby, and M. S. Sayed, "Low cost luma interpolation filter for motion compensation in HEVC," in *Proc. Int. Symp. Midwest Circuits Syst.*, Abu Dhabi, UAE, 2016, pp. 1–4.
[13] C. M. Diniz, M. Shafique, S. Bampi, and J. Henkel, "High-throughput interpolation hardware architecture with coarse-grained reconfigurable datapaths for HEVC," in *Proc. IEEE Int. Conf. Image Process.*, Melbourne, VIC, Australia, 2013, pp. 2091–2095.
[14] G. Pastuszak and M. Trochimiuk, "Architecture design and efficiency evaluation for the high-throughput interpolation in the HEVC encoder," in *Proc. Euromicro Conf. Digit. Syst. Design*, Los Alamitos, CA, USA, 2013, pp. 423–428.
[15] C.-Y. Lung and C.-A. Shen, "A high-throughput interpolator for fractional motion estimation in high efficient video coding (HEVC) systems," in *Proc. IEEE Asia–Pac. Conf. CAS*, Ishigaki, Japan, 2014, pp. 268–271.
[16] W. Zhou, X. Zhou, and X. Lian, "An efficient interpolation filter VLSI architecture for HEVC," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Brisbane, QLD, Australia, 2015, pp. 1106–1110.
[17] D. Kang, Y. Kang, and Y. Hong, "VLSI implementation of fractional motion estimation interpolation for high efficiency video coding," *Electron. Lett.*, vol. 51, no. 15, pp. 1163–1165, Jul. 2015.
[18] S. Wang, D. Zhou, J. Zhou, T. Yoshimura, and S. Goto, "VLSI implementation of HEVC motion compensation with distance biased direct cache mapping for 8K UHDTV applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 2, pp. 380–393, Feb. 2017.
[19] G. Pastuszak and M. Trochimiuk, "Algorithm and architecture design of the motion estimation for the H.265/HEVC 4K-UHD encoder," *J. Real Time Image Process.*, vol. 12, no. 2, pp. 517–529, Aug. 2016.
[20] C. M. Diniz, M. Shafique, S. Bampi, and J. Henkel, "A reconfigurable hardware architecture for fractional pixel interpolation in high efficiency video coding," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 2, pp. 238–251, Feb. 2015.
[21] K. McCann *et al.*, *High Efficiency Video Coding (HEVC) Test Model 15 (HM 15) Encoder Description*, document JCTVC-Q1002, JCTVC, Valencia, Spain, Jun. 2014.
[22] F. Bossen, *Common Test Conditions and Software Reference Configurations*, document JCTVC-I1100, JCTVC, Geneva, Switzerland, May 2012.

Ercan Kalali received the B.S. degree in electronics engineering from Istanbul Technical University, Istanbul, Turkey, in 2011, the M.S. degree in electronics engineering from Sabanci University, Istanbul, in 2013, where he is currently pursuing the Ph.D. degree. His research interests include low power digital hardware design for digital video processing and coding.

Ilker Hamzaoglu (M'00–SM'12) received the B.S. and M.S. degrees in computer engineering from Bogazici University, Istanbul, Turkey, in 1991 and 1993, respectively, and the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign, IL, USA, in 1999. He is currently as a Senior and Principle Staff Engineer with Multimedia Architecture Laboratory, Motorola Inc., Schaumburg, IL, USA, from 1999 and 2003. He is currently an Associate Professor with Sabanci University, Istanbul, where he has been a Faculty Member since 2003. His research interests include low power digital hardware design for video processing and compression, system-on-chip ASIC and FPGA design, embedded system design, computer-aided design and test for digital very large scale integration circuits.