A Reconfigurable Fractional Interpolation Hardware for VVC Motion Compensation

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

***Abstract*—Fractional interpolation is one of the most computationally complex parts of video compression standards. Fractional interpolation in Versatile Video Coding (VVC) standard has much higher computational complexity than fractional interpolation in previous video compression standards. In this paper, a reconfigurable VVC fractional interpolation hardware for motion compensation is designed and implemented using Verilog HDL. The proposed hardware is the first VVC fractional interpolation hardware for motion compensation in the literature. It interpolates necessary fractional pixels for 1/16 pixel accuracy for all prediction unit sizes. The proposed VVC fractional interpolation hardware, in the worst case, can process 66 quad full HD (3840x2160) frames per second. It has up to 77% less power consumption than baseline VVC fractional interpolation hardware.**

***Keywords*— VVC, motion compensation, fractional interpolation, hardware implementation, FPGA.**

I. INTRODUCTION

ITU and ISO are developing a new international video compression standard called Versatile Video Coding (VVC) [1]-[6]. VVC will have higher compression efficiency than High Efficiency Video Coding (HEVC) standard at the expense of much higher computational complexity [7]-[10]. One of the most computationally complex parts of VVC is fractional interpolation.

HEVC standard uses 3 different 8-tap FIR filters for fractional interpolations and provides 1/4 fractional pixel accuracy. However, VVC standard uses 15 different 8-tap FIR filters for fractional interpolations and provides 1/16 fractional pixel accuracy. Therefore, VVC fractional interpolation has much higher computational complexity than HEVC fractional interpolation.

In this paper, a reconfigurable VVC fractional interpolation hardware for motion compensation (MC) is proposed. The proposed hardware supports all prediction unit (PU) sizes. It interpolates necessary fractional pixels for the fractional pixel location in an 8x8 PU pointed by the given fractional pixel accurate motion vector. For larger PU sizes, the PU is divided into 8x8 blocks, and these blocks are interpolated separately. Since the proposed hardware is used for motion compensation stage of VVC encoder and decoder, only one fractional pixel per integer pixel is required. Therefore, the proposed hardware has a reconfigurable datapath which can be configured to implement any of the 15 different 8-tap FIR filters.

The proposed VVC fractional interpolation hardware is implemented using Verilog HDL. The Verilog RTL code is verified to work at 250 MHz on a Xilinx Virtex 7 FPGA. The proposed VVC fractional interpolation hardware, in the worst case, can process 66 quad full HD (3840x2160) frames per second. The proposed reconfigurability reduced the power consumption of FPGA implementation of the proposed VVC fractional interpolation hardware by 77%.

The proposed hardware is the first VVC fractional interpolation hardware for motion compensation in the literature. Several HEVC fractional interpolation hardware implementations are proposed in the literature [11]-[16]. In Section III, the VVC fractional interpolation hardware proposed in this paper is compared with them.

The rest of the paper is organized as follows. In Section II, VVC fractional interpolation algorithm is explained. In Section III, the proposed reconfigurable VVC fractional interpolation hardware is presented, and its implementation results are given. Finally, Section IV presents the conclusions.

II. VVC FRACTIONAL INTERPOLATION ALGORITHM

VVC standard uses 15 different 8-tap FIR filters for fractional pixel interpolation. The coefficients of these 15 FIR filters are shown in Table I. A_{-3} – A_4 show input pixels for a filter where sub-indices represent the indices of coefficients. The F_7 8-tap FIR filter equation is shown in (1) as an example.

$$F_7 = (-A_{-3} + 4*A_{-2} - 11*A_{-1} + 45*A_0 + 34*A_1 - 10*A_2 + 4*A_3 + A_4) >> 6 \quad (1)$$

Integer pixels, fractional pixels and FIR filters used to interpolate these fractional pixels are shown in Fig. 1. There are 255 fractional (half and quarter) pixels for one integer pixel. There are 15 half-pixels between two neighboring horizontal integer pixels called horizontal half-pixels. There are 15 half-pixels between two neighboring vertical integer pixels called vertical half-pixels. These 15 horizontal and 15 vertical half-pixels are interpolated from nearest integer pixels in horizontal and vertical directions, respectively, using 15 different 8-tap FIR filters. There are 15x15=225 quarter-pixels between 15 horizontal and 15 vertical half-pixels. These quarter-pixels are interpolated from nearest horizontal half-pixels using 15 different 8-tap FIR filters.

TABLE I. VVC FRACTIONAL INTERPOLATION FILTERS

Filters	Coefficients							
	A_{-3}	A_{-2}	A_{-1}	A_0	A_1	A_2	A_3	A_4
1	0	1	-3	63	4	-2	1	0
2	-1	2	-5	62	8	-3	1	0
3	-1	3	-8	60	13	-4	1	0
4	-1	4	-10	58	17	-5	1	0
5	-1	4	-11	52	26	-8	3	-1
6	-1	3	-9	47	31	-10	4	-1
7	-1	4	-11	45	34	-10	4	-1
8	-1	4	-11	40	40	-11	4	-1
9	-1	4	-10	34	45	-11	4	-1
10	-1	4	-10	31	47	-9	3	-1
11	-1	3	-8	26	52	-11	4	-1
12	0	1	-5	17	58	-10	4	-1
13	0	1	-4	13	60	-8	3	-1
14	0	1	-3	8	62	-5	2	-1
15	0	1	-2	4	63	-3	1	0

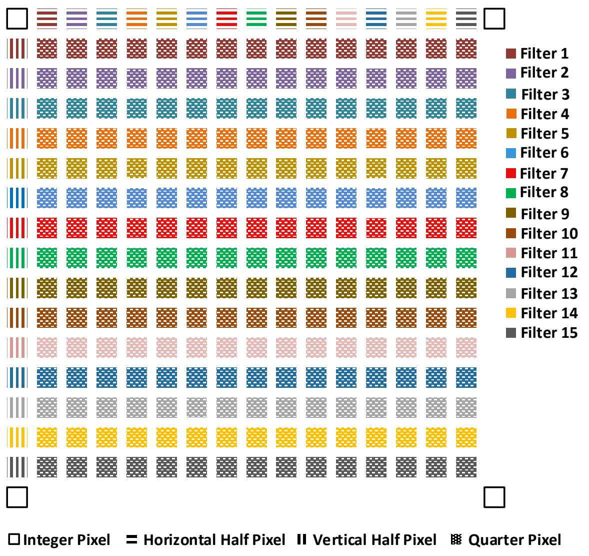

Fig. 1. Integer, half and quarter pixels.

VVC fractional interpolation algorithm used for motion compensation interpolates necessary fractional pixels for one out of 255 fractional pixel locations pointed by the given 1/16 pixel accurate motion vector. Necessary fractional pixels are determined using x fraction and y fraction of the given 1/16 pixel accurate motion vector. If either x fraction or y fraction is zero, only necessary half-pixels are interpolated. If neither x fraction nor y fraction is zero, horizontal half-pixels necessary to interpolate the quarter-pixels are interpolated first. Then, the necessary quarter-pixels are interpolated using these horizontal half-pixels.

III. PROPOSED VVC FRACTIONAL INTERPOLATION HARDWARE

The proposed reconfigurable VVC fractional interpolation hardware for all PU sizes is shown in Fig. 2. The proposed hardware interpolates the necessary fractional pixels for luma component of an 8x8 PU for a given 1/16 pixel accurate motion vector using integer or half-pixels. For larger PU sizes, the PU is divided into 8x8 blocks and these blocks are interpolated separately. For example, a 16x16 PU is divided into four 8x8 blocks and each 8x8 block is interpolated separately.

Since 15x8 horizontal half-pixels are necessary for interpolating quarter-pixels, 15x8x8 on-chip transpose memory is used to store horizontal half-pixels necessary for interpolating quarter-pixels in certain cases. The horizontal half-pixels interpolated from nearest integer pixels in horizontal direction are stored in transpose memory horizontally in 15 clock cycles. Then, 15 horizontal half-pixels are read vertically from transpose memory in each clock cycle to interpolate quarter-pixels.

The proposed hardware takes 15 integer pixels in each clock cycle. It interpolates 8 fractional pixels in each clock cycle using 8 parallel reconfigurable datapaths. If the necessary fractional pixels are half-pixels, 8x8 half-pixels are interpolated using the integer pixels in 8 clock cycles. If the necessary fractional pixels are quarter-pixels, 15x8 horizontal half-pixels are interpolated using the integer pixels in 15 clock cycles. Then, 8x8 quarter-pixels are interpolated using these horizontal half-pixels in 8 clock cycles. There are three pipeline stages in the proposed hardware. Therefore, the proposed hardware interpolates the half-pixels and quarter-pixels for an 8x8 PU in 11 and 29 clock cycles, respectively.

15 different 8-tap FIR filters are used to interpolate half-pixels and quarter-pixels. Last 7 FIR filters are symmetric of the first 7 FIR filters. Therefore, in this paper, a reconfigurable datapath which implements the first 8 FIR filters is proposed. It can be configured to calculate output of any of the first 8 FIR filters. To calculate output of one of the last 7 FIR filters using the proposed reconfigurable datapath, inputs are reversed and corresponding symmetric filter is selected.

The proposed reconfigurable datapath is shown in Fig. 3. It implements multiplications with constant coefficients using adders and shifters. It has 14 adders/subtractors and their inputs are determined by a filter selection signal. It selects different input pixels with different shift amounts for each fractional interpolation equation using input multiplexers as shown in Table II.

In this paper, a baseline VVC fractional interpolation hardware is also designed and implemented for comparison. The baseline hardware has the same architecture as the proposed hardware. The only difference is their datapaths. In the baseline hardware datapath, all 15 FIR filters are implemented separately and output of one FIR filter is selected based on filter selection signal. Therefore, the baseline hardware datapath has 91 adders while the proposed reconfigurable datapath has 14 adders.

The proposed and the baseline VVC fractional interpolation hardware are implemented using Verilog HDL. The Verilog RTL codes are verified with RTL simulations. The Verilog RTL codes are synthesized and mapped to a Xilinx VC7VX330T-3FFG1157 FPGA using Xilinx ISE 14.7. The FPGA implementations are verified with post place and route simulations. The simulation results matched the results of a software implementation of VVC fractional interpolation algorithm.

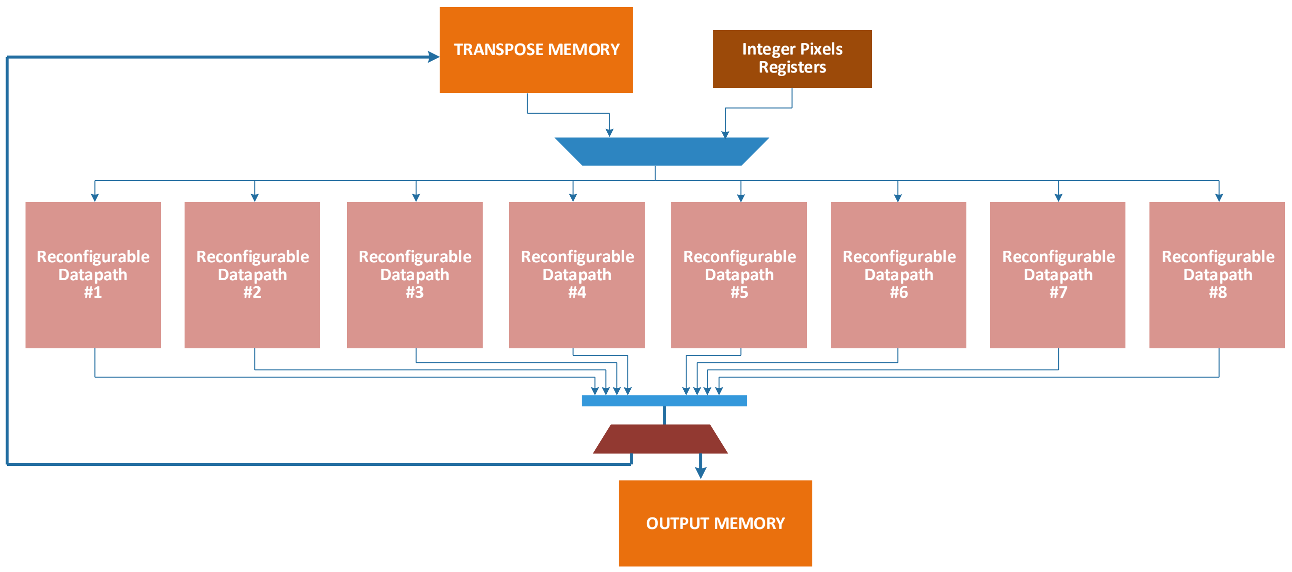
TRANSPOSE MEMORY
Integer Pixels Registers
Reconfigurable Datapath #1
Reconfigurable Datapath #2
Reconfigurable Datapath #3
Reconfigurable Datapath #4
Reconfigurable Datapath #5
Reconfigurable Datapath #6
Reconfigurable Datapath #7
Reconfigurable Datapath #8
OUTPUT MEMORY

Fig. 2. Proposed VVC fractional interpolation hardware.

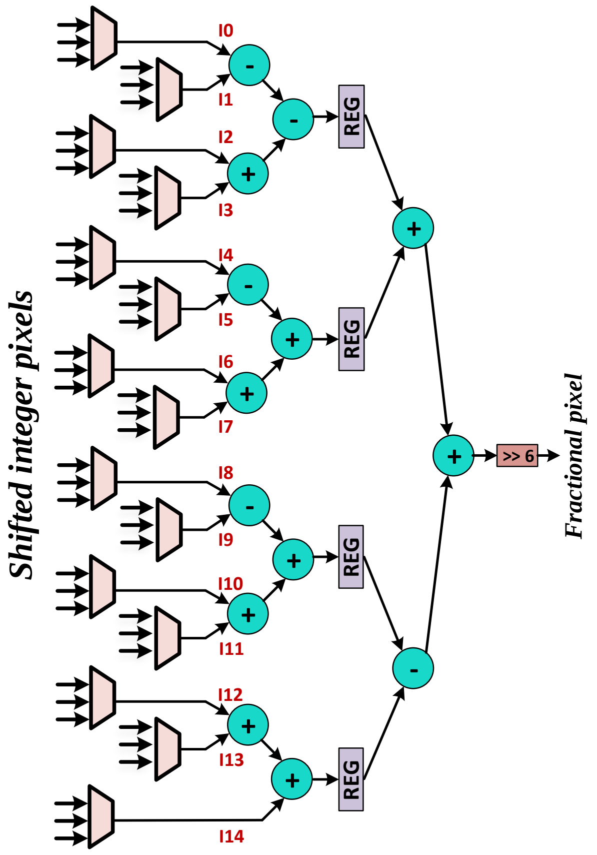

Fig. 3. Proposed reconfigurable datapath.

As shown in Fig. 4, FPGA implementations are also verified to work correctly on a Xilinx Virtex 7 VC707 FPGA board which includes an FPGA, 1 GB DRAM and interfaces such as UART and HDMI. Microblaze processor reads video frames from computer, stores them to DDR memory and sends them to FPGA using high-speed AXI-4 bus. The proposed hardware interpolates the video frames. Then, interpolated video frames are displayed on HDMI monitor.

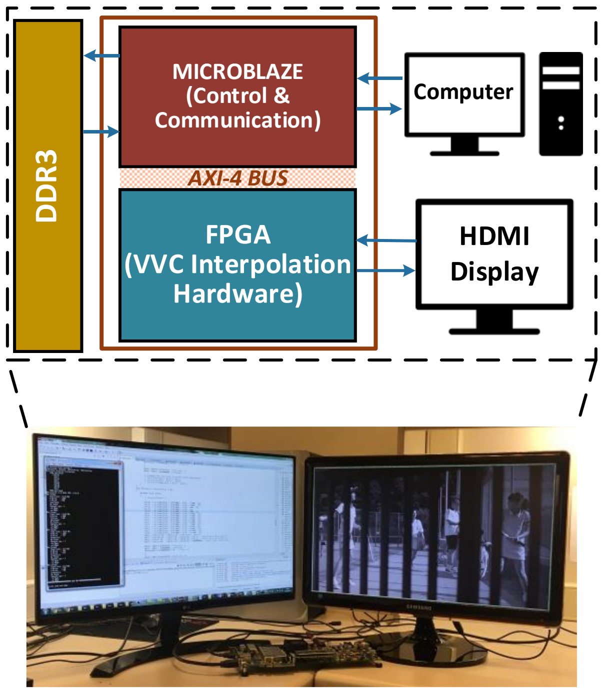
DDR3
MICROBLAZE (Control & Communication)
AXI-4 BUS
FPGA (VVC Interpolation Hardware)
Computer
HDMI Display

Fig. 4. FPGA board implementation.

As shown in Table III, FPGA implementation of the proposed VVC fractional interpolation hardware uses 1688 DFFs and 4467 LUTs. It can work at 250 MHz, and it can process 66 quad full HD (3840x2160) frames per second. FPGA implementation of the baseline VVC fractional interpolation hardware uses 5446 DFFs and 12016 LUTs. It can work at 238 MHz, and it can process 63 quad full HD (3840x2160) frames per second.

The Verilog RTL codes of the baseline and proposed VVC fractional interpolation hardware are also synthesized to TSMC 90 nm standard cell library, and the resulting netlists are placed and routed. As shown in Table III, ASIC implementations of the baseline and proposed hardware use

TABLE II. RECONFIGURABLE DATAPATH INPUTS

Filters	I0	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12	I13	I14
1	0	0	B	0	C<<2	D<<6	C	D	E<<2	0	0	F<<1	0	G	0
2	A	B<<1	0	C<<2	C<<1	D<<6	C	D<<1	E<<3	0	0	F<<1	F	G	0
3	A	B<<1	B	C<<3	0	D<<6	0	D<<2	E<<4	E	E<<2	F<<2	0	G	0
4	A	B<<2	0	C<<3	C<<1	D<<6	D<<1	D<<3	E<<4	E	0	F<<2	F	G	0
5	A	B<<2	C	C<<3	C<<2	D<<6	D<<2	D<<4	E<<5	E<<1	E<<3	F<<3	G	G<<2	H
6	A	B<<1	B	C<<3	C<<1	D<<5	D<<4	D	E<<5	C	E	F<<3	F<<1	G<<2	H
7	A	B<<2	D	C<<3	C<<1	D<<5	D<<4	D<<2	E<<5	E<<1	C	F<<3	F<<1	G<<2	H
8	A	B<<2	F	C<<3	C<<1	D<<5	D<<3	0	E<<5	E<<3	C	F<<3	F<<2	G<<2	H

TABLE IV. HARDWARE COMPARISON

	[11]	**[12]**	**[13]**	**[14]**	**[15]**	**[16]**	**Proposed**
FPGA	Xilinx Virtex 6	Xilinx Virtex 6	Arria II GX	Xilinx Virtex 5	Stratix III	Xilinx Virtex 6	Xilinx Virtex 7
Slices	---	---	---	2181	---	1498	1407
LUTs	3005	3929	18831	5017	7701	3806	4467
Block RAMs	2	6	---	2	---	---	---
Max. Freq. (MHz)	100	200	200	283	278	233	250
Frames per Second	64 2560x1600	30 3840x2160	60 1920x1080	30 2560x1600	60 3840x2160	35 3840x2160	66 3840x2160
Design	Only MC	ME + MC	ME + MC	ME + MC	ME + MC	ME + MC	Only MC
Standard	HEVC	HEVC	HEVC	HEVC	HEVC	HEVC	VVC

TABLE III. IMPLEMENTATION RESULTS

	Baseline		**Proposed**	
Technology	Xilinx Virtex 7	TSMC 90 nm	Xilinx Virtex 7	TSMC 90 nm
Slice/Gate Count	3630	48.3 K	1407	11.7 K
DFF	5446	---	1688	---
LUT	12016	---	4467	---
Max. Freq. (MHz)	238	417	250	357
Frames per Second	63 3840x2160	110 3840x2160	66 3840x2160	95 3840x2160

TABLE V. POWER CONSUMPTION RESULTS

	Baseline		**Proposed**	
Frame	**Tennis**	**Kimono**	**Tennis**	**Kimono**
Clock (mW)	68.33	68.33	13.84	13.84
Signal (mW)	96.75	131.64	16.64	22.58
Logic (mW)	99.27	135.36	32.40	40.64
Total Power (mW)	264.35	335.33	62.88	77.06
Power Reduction	---	---	76.21 %	77.02 %

48.3K and 11.7K gates, respectively, based on NAND (2x1) gate area excluding on-chip memory. ASIC implementations of the baseline and proposed hardware can work at 417 and 357 MHz, respectively, and they can process 110 and 95 quad full HD frames per second, respectively.

Since the proposed hardware is the first VVC fractional interpolation hardware for motion compensation in the literature, it is compared with HEVC fractional interpolation hardware in the literature [11]-[16]. The comparison is shown in Table IV. The HEVC fractional interpolation hardware proposed in [11] is designed for motion compensation. The others can be used for both motion estimation (ME) and motion compensation.

Since VVC fractional interpolation has higher computational complexity than HEVC fractional interpolation, the proposed hardware has higher area than the HEVC fractional interpolation hardware proposed in [11]. However, since the proposed hardware is designed for motion compensation, it does not have higher area than the other HEVC fractional interpolation hardware in the literature.

Power consumptions of the baseline and proposed hardware are estimated using Xilinx XPower Analyzer tool. Post place and route timing simulations are performed for Tennis and Kimono (1920x1080) video frames at 100 MHz [17]. The signal activities of these timing simulations are stored in VCD files, and they are used for estimating the power consumptions of FPGA implementations. The power consumptions of both the baseline and proposed hardware are shown in Table V. Clock, signal and logic power consumptions are given for detailed analysis. Total power consumption of the proposed hardware for Tennis and Kimono frames is 76.21% and 77.02% less than that of the baseline hardware, respectively.

IV. Conclusion

In this paper, a reconfigurable VVC fractional interpolation hardware for all PU sizes is proposed. It is the first fractional interpolation hardware for VVC motion compensation in the literature. The proposed VVC fractional interpolation hardware can process 66 quad full HD (3840x2160) frames per second on a Xilinx Virtex 7 FPGA. It has up to 77% less power consumption than baseline VVC fractional interpolation hardware on the same FPGA.

References

[1] J. Chen, Y. Chen, M. Karczewiz, X. Li, H. Liu, L. Zhang, and X. Zhao, "Coding tools investigation for next generation video coding," ITU-T SG16 COM16–C806, Feb. 2015.

[2] J. Chen, E. Alshina, G. J. Sullivan, J. R. Ohm, and J. Boyce, "Algorithm description of joint exploration model 7," JVET-G1001, July 2017.

[3] H. Azgin, A. C. Mert, E. Kalali, and I. Hamzaoglu, "Reconfigurable intra prediction hardware for future video coding," IEEE Trans. on Consumer Electronics, vol. 63, no. 4, pp. 419-425, Nov. 2017.

[4] A. C. Mert, E. Kalali, and I. Hamzaoglu, " High performance 2D transform hardware for future video coding," IEEE Trans. on Consumer Electronics, vol. 63, no. 2, pp. 117-125, May 2017.

[5] A. C. Mert, E. Kalali, and I. Hamzaoglu, "An FPGA implementation of future video coding 2D transform," IEEE Int. Conf. on Consumer Electronics – Berlin (ICCE-Berlin), pp. 31-36, Sep. 2017.

[6] M. J. Garrido, F. Pescador, M. Chavarrias, P. J. Lobo, and C. Sanz, "A high performance FPGA-based architecture for the future video coding adaptive multiple core transform," IEEE Trans. on Consumer Electronics, vol. 64, no. 1, pp. 53-60, Feb. 2018.

[7] J. Vanne, M. Viitanen, T.D. Hämäläinen, and A. Hallapuro, "Comparative rate-distortion-complexity analysis of HEVC and AVC video codecs", IEEE Trans. on Circuits and Systems for Video Technology, vol. 22, no. 12, pp.1885-1898, Dec. 2012.

[8] E. Kalali, Y. Adibelli, and I. Hamzaoglu, "A high performance and low energy intra prediction hardware for high efficiency video coding," Int. Conf. on Field Programmable Logic and Applications (FPL), pp. 719-722, Aug. 2012.

[9] E. Kalali, E. Ozcan, O. M. Yalcinkaya, and I. Hamzaoglu, "A low energy HEVC inverse transform," IEEE Trans. on Consumer Electronics, vol. 60, no. 4, pp. 754-761, Nov. 2014.

[10] E. Kalali, A. C. Mert, and 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.

[11] E. Kalali, Y. Adibelli, and I. Hamzaoglu, "A reconfigurable HEVC sub-pixel interpolation hardware", IEEE Int. Conference on Consumer Electronics - Berlin, Sept. 2013.

[12] E. Kalali and I. Hamzaoglu, "A low energy HEVC sub-pixel interpolation hardware," IEEE Int. Conference on Image Processing, pp. 1218-1222, Oct. 2014.

[13] G. Pastuszak and M. Trochimiuk, "Architecture design and efficiency evaluation for the high-throughput interpolation in the HEVC encoder", 16th Euromicro Conference on Digital System Design, Sep. 2013.

[14] 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. on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 2, pp. 238-251, Feb. 2015.

[15] H. Maich, C. Afonso, D. Franco, B. Zatt, M. Porto, and L. Agostini, "High throughput hardware design for the HEVC fractional motion estmation interpolation unit", IEEE 20th International Conference on Electronics, Circuits, and Systems, May 2014.

[16] A. C. Mert, E. Kalali, and I. Hamzaoglu, "An HEVC fractional interpolation hardware using memory based constant multiplication," IEEE Int. Conf. on Consumer Electronics (ICCE), pp. 742-746, Jan. 2018.

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