FPGA-ACCELERATED HEVC ENCODER FOR ENERGY-EFFICIENT MULTI-ACCESS EDGE COMPUTING

Panu Sjövall, Alexandre Mercat, and Jarno Vanne

Ultra Video Group, Tampere University, Finland

ABSTRACT

High Efficiency Video Coding (HEVC) and Multi-access Edge Computing (MEC) technologies can make real-time streaming media services available to users with reasonable bandwidth, but the computational complexity of HEVC tends to lead to increased energy consumption in these schemes. In this paper, we investigate the energy saving opportunities of utilizing a field-programmable gate array (FPGA) based HEVC encoder in edge media servers and devices. In practice, we analyze the energy impact of migrating our Kvazaar software HEVC intra encoder to Intel Arria 10 PCIe FPGA(s) on two platforms: 1) Nokia Airframe Cloud Server with 2.4 GHz dual 14-core Intel Xeon processors and 2) an embedded Jetson AGX Orin board with 2.2 GHz 12-core ARM processor. According to our experiments, FPGA encoding on these two platforms saved 76% and 86% of the energy taken up by software only encoding on Airframe, respectively. These results indicate the potential of FPGA-based video encoder acceleration in future green MEC architectures.

Index Terms—*High Efficiency Video Coding (HEVC/H.265), Kvazaar HEVC encoder, field-programmable gate array (FPGA), energy efficiency, Multi-access Edge Computing (MEC)*

1 INTRODUCTION

The proliferation of multifaceted media applications has set stringent performance requirements for modern embedded media devices. To that end, edge computing techniques are increasingly deployed to offload the most compute-intensive tasks from end devices to external computing resources in the cloud. However, it is of utmost importance for many applications that the external computing capacity is in close proximity to the user. *European Telecommunications Standards Institute* (*ETSI*) has addressed this need by specifying a network architecture concept called *Multi-access Edge Computing* (*MEC*) [1] that moves the computing and storage resources from a centralized cloud to the network edge. MEC architectures can offer low latency and high bandwidth, which are key enabling factors for many real-time streaming media applications [2].

Efficient video coding formats are essential to make the most of the bandwidth provided by MEC architectures. Over the past few decades, ISO/IEC MPEG and ITU-T VCEG have released a series of international video coding standards. These include the widely used *Advanced Video Coding* (*AVC/H.264*) [3], the well-established *High Efficiency Video Coding* (*HEVC/H.265*) [4], and emerging *Versatile Video Coding* (*VVC/H.266*) [5]. This work focuses on HEVC, one of the most widespread video formats currently [6].

This paper is part of the ADACORSA project that has received funding within the ECSEL JU in collaboration with the European Union's H2020 Framework Programme (H2020/2014-2020) and National Authorities, under grant agreement 876019, and the Academy of Finland (decision no. 349216).

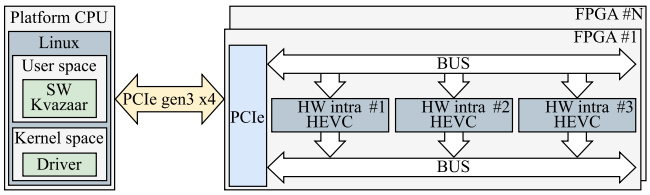

Fig. 1. Overview of the FPGA-accelerated HEVC intra encoder.

HEVC is able to attain high coding efficiency, but its enormous computational complexity tends to escalate the energy consumption in different MEC schemes. Maximizing battery life on energy-constrained embedded devices and minimizing the power consumption of servers require power-efficient implementations and optimization techniques.

A couple of previous works have explored energy consumption of video encoders and proposed energy saving techniques for them. The encoder-level evaluations included the HEVC refence software *HEVC test model* (*HM*) [7], [8], AVC reference software *Joint Test Model* (*JM*) [8], and various open-source *software* (*SW*) video encoders [9]–[12]. In addition, some alternative energy models were introduced to the HM [13] and x265 [14], [15] encoders. Furthermore, several energy-efficient coding tool implementations were proposed on *field-programmable gate array* (*FPGA*) and *application specific integrated circuit* (*ASIC*), including intra prediction, sub-pixel interpolation, *discrete cosine transform* (*DCT*), *inverse-DCT* (*IDCT*), and *Sum of Absolute Transformed Differences* (*SATD*) [16]–[22]. Nevertheless, to the best of our knowledge, none of these works have evaluated the energy reduction obtained by migrating the entire encoder to FPGA.

This paper proposes to save energy of MEC architectures by replacing SW HEVC intra encoding with customizable *hardware* (*HW*) HEVC intra encoder on FPGA [23]. This approach, illustrated in Fig. 1, allows for offloading the most demanding encoding tools from *central processing unit* (*CPU*) to FPGA. To evaluate the effectiveness of this approach, the open-source HEVC encoder Kvazaar [24], [25] is used as an anchor for our energy evaluation. The experiments are conducted on a dual Xeon CPU (x86) server and a Jetson Orin board (ARM), with the server utilizing up to two PCIe FPGAs for HW encoding and the Jetson using one. The energy consumption is measured with two encoding configurations designed for practical live encoding scenarios. The results are used to compare the energy efficiency of the proposed FPGA-accelerated HEVC video encoder with the SW-only solutions.

The remainder of the paper is organized as follows. Section II introduces our FPGA implementation used to replace SW Kvazaar HEVC intra encoder. The experimental setup is described in Section III. Sections IV and V present the initial power analysis and energy saving results, respectively. Finally, Section VI concludes the paper.

Table 1. Profiled edge server and edge device platforms.

Platform	Nokia Airframe	Jetson AGX Orin
Processor	2× Intel Xeon E5-2680 v4 (14×2.4 GHz)	Arm Cortex-A78AE v8.2 (12×2.2 GHz)
Architecture	x86 / AVX2	ARM / Neon
Memory	256 GB	64 GB
Compiler	GCC 7.3.1	GCC 9.4.0
Operating system	CentOS 6.9	Ubuntu 20.04
FPGA	Max. 2× PCIe FPGA (Intel Arria 10 GX 1150)	Max. 1× PCIe FPGA (Intel Arria 10 GX 1150)

Table 2. Test sequences.

Dataset	Resolution	Name(s)	Frame rate	Number of frames	
				Single	Loop
UVG	3840×2160	SunBath	50	300	1800
		CityAlley, FlowerFocus, FlowerKids, FlowerPan, RaceNight, RiverBank, Twilight	50	600	3600
CTC-A	2560×1600	PeopleOnStreet, Traffic	30	150	3600
CTC-B	1920×1080	BasketballDrive, Cactus	50	500	12000
		BQTerrace	60	600	14400
		Kimono, ParkScene	24	240	5760

2 FPGA-ACCELERATED HEVC ENCODING

This work is based on our Kvazaar open-source HEVC encoder [24], [25] and its corresponding intra encoder implementation on FPGA [23]. In Kvazaar, individual coding tools are written in C/C++. For the FPGA [23] implementation, intra coding and arithmetic coding tools have also been optimized for *High-level Synthesis* (*HLS*) [26], and the *register-transfer level* (*RTL*) description has been automatically generated from the HLS code with the Catapult HLS tool [27].

Fig. 1 illustrates the FPGA-accelerated HEVC intra encoder, which is partitioned between a server CPU and one or more FPGAs. Each FPGA is connected to the server via a *PCI Express* (*PCIe*) gen3 x4 bus.

On the server side, a Kvazaar SW instance runs in the user space and takes care of 1) raw video input; 2) HEVC stream initialization; 3) parallelization of *coding tree units* (*CTUs*); 4) offloading the intra coding task to FPGAs; and 5) receiving the encoded CTU bitstream and related parameters from FPGA. The connection between the Kvazaar SW instance and the FPGA(s) is enabled by a dedicated Linux driver in the kernel space. This approach allows for the utilization of practically arbitrary number of parallel PCIe FPGAs.

On each FPGA, the number of HW intra encoder instances depends on the available logic area. Each instance includes all necessary coding tools for all-intra coding. The FPGA used in this work has the capacity for three parallel instances [23], each supporting parallel processing of 16 individual CTUs [4]. This results in a total of up to 48 CTUs processed in parallel per FPGA.

For a comprehensive understanding of this implementation, please refer to our previous work [23].

3 EXPERIMENTAL SETUP

The proposed FPGA-accelerated HEVC intra encoder was evaluated on two platforms specified in Table 1: 1) the Nokia Airframe edge server and 2) the Jetson AGX Orin edge device. The Airframe platform has two 14-core Intel Xeon processors, and it can be equipped with up to two PCIe FPGA accelerator cards. Jetson, on the other hand, is a more compact platform with an option for a single FPGA. The FPGA chip used in the evaluation was an Intel Arria 10 10AX115S2F40E2SG on Intel Arria 10 GX FPGA board, which supports both PCIe generation 3 x4 and 40GbE fiber connections. In this study, a platform configuration with zero FPGAs refers to a SW-only solution, where the encoder was running solely on CPU of the tested platform. This SW-only configuration was used as an anchor in our experiments.

Two real-time encoding configurations, a.k.a., Kvazaar presets [25], *Fast* and *Ultrafast*, were used in our experiments. The former supports the blocks sizes of 8×8, 16×16, and 32×32, whereas the latter only tests 8×8 and 16×16 blocks. The experiments were conducted under the *All Intra* (*AI*) HEVC *common test conditions* (*CTC*) [28] condition with *quantization parameters* (*QPs*) values of 22, 27, 32, and 37. The encoding process was parallelized using tiles [4] along with *overlapped wavefront* (*OWP*) [29] to enable parallel processing of large number of CTUs. The test set consisted of 15 video sequences, including eight 4K sequences from the UVG dataset [30] and class A and B sequences from the HEVC CTC [28], as summarized in Table 2.

The power usage was evaluated using Fluke 1748 with clamps measuring the 1-phase socket. It was directly attached to the power supply of the encoding platform. This approach allowed a continuous power measurement with a sampling rate of 1 Hz and provided an accurate representation of the power consumption during encoding. The captured data was used to calculate the energy consumption in *kilojoules* (*kJ*). The accuracy of the energy consumption measurement was improved with two techniques. First, the number of tested frames was increased by looping the sequences of the UVG datasets 6 times and the CTC sequences 24 times; the rightmost column of Table 2 indicates the total number of frames encoded for each sequence. Secondly, each test run was conducted multiple times, and the results from individual runs were averaged to minimize the measurement error. The relative standard deviation between measurements was 0.9% for the Airframe and 0.3% for the Jetson.

The coding speed is also reported in *frames per second* (*fps*). Evaluating a trade-off between energy and coding speed provides insights into the overall performance of the proposed FPGA-accelerated HEVC encoder implementation. The coding efficiency of our FPGA-accelerated HEVC encoder is similar to that of the SW-only solution in terms of *Bjøntegaard Delta rate* (*BD-rate*) [31]. For reference, HM yields a 19% better BD-rate than the proposed system in AI coding [23].

4 POWER ANALYSIS

In the previous work, we optimized the performance/area ratio of our HW encoder [23]. This paper shifts our focus to energy efficiency by exploring the benefits of offloading demanding intra coding tools from CPU to dedicated HW in MEC architectures. Before conducting full-scale energy consumption analysis, we carried out an initial power analysis with our experimental setup to test the accuracy of the measuring tool and to identify the basic power characteristics of the two platforms with varying number of FPGAs.

Fig. 2 illustrates the power curves obtained for the CityAlley sequence with QP value of 22 and the *Fast* configuration on five different platform setups. Similar results were obtained with the other sequences, QP values, and *Ultrafast* configuration. The curves report the power input measured for the entire platform during idling and encoding. The same tests were run on each platform

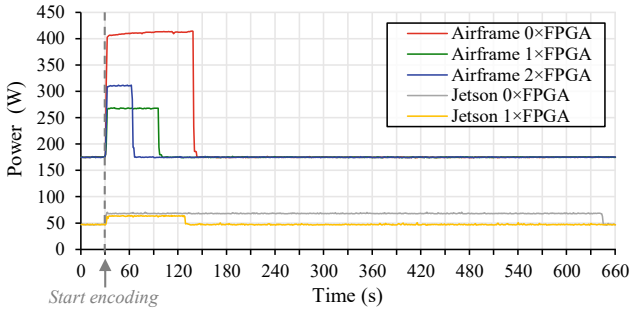

Fig. 2. Power input of five platform setups (CityAlley, QP 22, *Fast*).

configuration individually, and the results are interleaved in Fig. 2 so that the execution starts precisely at the 30-second mark. The results show that the power input measurement exhibits low levels of noise, indicating the high precision of the measuring tool. Moreover, Fig. 2 illustrates the power characteristics of the encoding process on the two platforms, including the idle and maximum power consumption as well as the total encoding time. Through analysis of the results, two potential energy-saving opportunities are identified: 1) reducing the maximum power input and 2) decreasing the encoding time.

One part of the energy savings comes from releasing the CPU from the complex encoding process [32]. As the CPU is only used for data transfer and control purposes, the threads are mostly idle when waiting for data from FPGA. Although CPU utilization is smaller with the FPGA-accelerated HEVC encoder over that of the SW-only coding, some power is still consumed. The PCIe FPGA used in this study is powered directly from the PCIe lane and has a maximum power input rating of 75 W. The energy consumption of the implemented intra encoder instance on FPGA was not measured individually. The idle power usage of the Airframe and Jetson platforms are 175 W and 47 W, respectively.

Table 3. Maximum power and speed (CityAlley, QP 22, *Fast*).

Platform	Number of FPGAs	Max. power (W)		Encoding speed (fps)	
		Abs.	Rel.	Abs.	Rel.
Airframe	0	414	1×	33	1×
	1	269	0.65×	56	1.7×
	2	312	0.75×	112	3.4×
Jetson	0	70	1×	6	1×
	1	64	0.91×	37	6.2×

Table 3 presents both absolute and relative maximum power inputs and encoding speeds for the experiments depicted in Fig. 2. According to our results, using one or two FPGAs with the Airframe platform reduces the maximum power input by 0.65× and 0.75×, respectively. On the other hand, incorporating FPGA into the Jetson platform results in a limited 0.91× reduction in the maximum power input. Even when counting in the idle power of the platform and the power rating of the FPGA, the proposed FPGA-accelerated HEVC encoder still provides a power reduction for the more energy-efficient Jetson platform. These results suggest that the power input of the FPGAs is less than the maximum rating.

The larger portion of the energy saving stems from the substantial increase in encoding speed achieved through FPGA

Table 4. Energy consumption and coding speed results for evaluated encoder configurations, platforms, and number of FPGAs.

		Config.	Energy Consumption (kJ)										Encoding Speed (fps)									
			Fast					*Ultrafast*					*Fast*					*Ultrafast*				
		Platform	Airframe			Jetson		Airframe			Jetson		Airframe			Jetson		Airframe			Jetson	
		Number of FPGAs	0	1	2	0	1	0	1	2	0	1	0	1	2	0	1	0	1	2	0	1
UVG (3840×2160)	CityAlley	QP 22	44.1	17.2	10.0	41.8	6.2	39.6	17.2	10.0	34.8	5.5	33	56	112	6	37	38	74	139	7	42
		QP 37	26.5	11.7	7.2	25.6	4.2	25.5	11.7	7.2	22.9	4.2	57	87	170	10	57	59	114	188	11	57
	FlowerFocus	QP 22	46.9	17.4	10.1	45.6	6.2	42.6	17.4	10.1	38.6	5.5	32	56	112	5	37	35	73	138	6	42
		QP 37	23.6	10.6	6.5	23.8	3.8	23.6	10.6	6.5	22.2	4.0	64	100	194	10	64	64	115	201	11	61
	FlowerKids	QP 22	40.6	16.7	9.7	38.7	6.0	37.2	16.7	9.7	32.9	5.5	36	58	116	6	38	40	73	134	7	42
		QP 37	30.4	12.5	7.6	28.7	4.5	28.6	12.5	7.6	25.1	4.4	49	81	157	9	53	53	108	202	10	55
	FlowerPan	QP 22	50.9	18.5	11.8	47.6	7.4	45.5	18.5	11.8	39.6	6.6	29	47	94	5	31	33	48	94	6	35
		QP 37	44.2	15.8	9.4	40.0	5.5	38.9	15.8	9.4	32.7	4.9	34	62	123	6	42	38	88	153	8	49
	RaceNight	QP 22	49.8	17.5	11.0	47.6	7.0	44.6	17.5	11.0	39.8	6.3	30	50	100	5	33	33	51	100	6	37
		QP 37	28.5	12.0	7.5	27.5	4.3	27.3	12.0	7.5	24.7	4.2	53	84	161	9	56	55	109	191	10	57
	RiverBank	QP 22	45.6	18.0	10.7	43.2	6.7	40.8	18.0	10.7	35.7	6.1	32	52	103	6	34	36	59	115	7	38
		QP 37	35.6	13.6	8.3	33.4	4.8	32.8	13.6	8.3	28.8	4.6	42	73	140	7	49	46	97	183	9	52
	SunBath	QP 22	18.3	7.6	4.6	17.6	2.7	16.6	7.6	4.6	14.9	2.4	40	65	128	7	43	45	89	157	8	48
		QP 37	12.9	5.9	3.7	12.4	1.9	12.4	5.9	3.7	11.2	1.9	58	86	165	10	59	60	113	196	11	64
	Twilight	QP 22	44.2	17.1	9.9	42.2	6.0	39.5	17.1	9.9	35.2	5.4	33	56	112	6	38	38	75	134	7	43
		QP 37	22.2	10.8	6.8	22.0	3.9	22.8	10.8	6.8	21.0	4.0	67	97	185	11	61	66	115	201	12	61
Class A (2560×1600)	PeopleOnStreet	QP 22	24.2	9.1	5.8	22.5	3.6	21.5	9.1	5.8	18.3	3.1	60	96	193	11	63	68	111	215	13	74
		QP 37	22.0	8.3	4.9	19.2	2.8	18.9	8.3	4.9	15.4	2.4	67	118	236	13	81	78	164	299	16	96
	Traffic	QP 22	24.6	9.6	5.9	23.0	3.7	21.9	9.6	5.9	18.7	3.3	60	95	191	11	62	67	106	206	13	71
		QP 37	21.9	8.0	4.7	19.5	2.7	19.2	8.0	4.7	15.8	2.3	68	122	241	12	84	78	170	321	15	99
Class B (1920×1080)	BasketballDrive	QP 22	41.5	15.8	10.1	39.2	6.4	37.2	15.8	10.1	32.6	5.6	118	182	364	21	119	132	198	384	26	137
		QP 37	32.6	12.3	7.4	30.3	4.3	29.1	12.3	7.4	25.1	3.9	152	267	523	28	181	171	375	623	33	203
	BQTerrace	QP 22	52.1	22.5	14.3	47.7	8.3	46.2	22.5	14.3	38.6	7.5	112	152	297	21	107	127	154	298	26	120
		QP 37	42.2	15.6	9.3	37.7	5.6	37.0	15.6	9.3	30.9	5.1	140	254	502	27	168	160	346	638	33	188
	Cactus	QP 22	43.5	16.1	10.4	40.6	6.4	39.2	16.1	10.4	34.1	5.8	112	175	346	20	117	125	175	342	24	132
		QP 37	36.1	13.0	7.8	32.7	4.7	31.7	13.0	7.8	26.7	4.2	137	251	499	25	168	156	349	651	31	191
	Kimono	QP 22	19.0	7.3	4.3	18.3	2.7	17.0	7.3	4.3	15.0	2.3	124	210	419	22	137	138	263	506	26	159
		QP 37	15.4	6.2	3.6	14.2	1.9	14.1	6.2	3.6	12.0	1.8	153	267	525	27	185	168	373	615	32	211
	ParkScene	QP 22	20.5	7.7	5.0	18.7	3.0	18.6	7.7	5.0	15.6	2.6	114	181	358	21	120	126	186	362	25	134
		QP 37	17.5	6.1	3.7	15.5	2.1	15.5	6.1	3.7	12.7	1.9	135	251	496	25	171	152	344	623	31	196

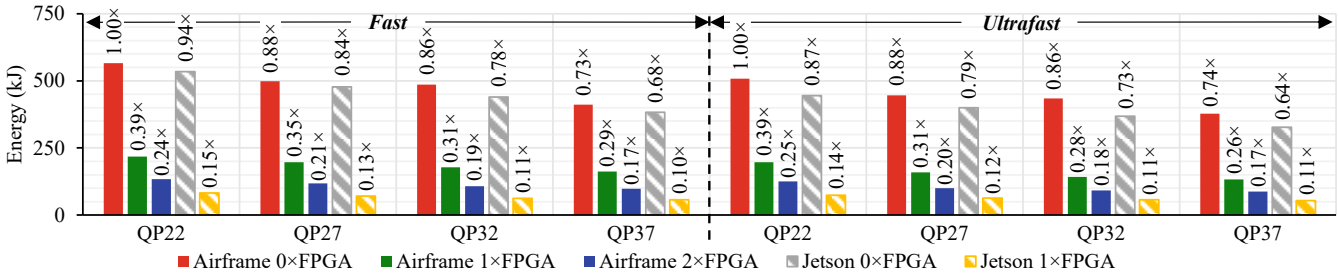

Fig. 3. Aggregated energy consumption results for the evaluated encoder configurations, platforms, and number of FPGAs.

Table 5. Energy consumption and average encoding speed.

			Platform	Airframe			Jetson	
			Number of FPGAs	0	1	2	0	1
Fast configuration	Relative energy consumption		UVG	1×	0.40×	0.24×	0.95×	0.14×
			class A	1×	0.37×	0.23×	0.90×	0.13×
			class B	1×	0.37×	0.23×	0.92×	0.13×
			All	**1×**	**0.39×**	**0.24×**	**0.93×**	**0.14×**
	Encoding speed	fps	UVG	42	69	135	7	46
			class A	63	107	214	11	72
			class B	128	222	440	23	148
		speedup	UVG	1×	1.65×	3.24×	0.17×	1.10×
			class A	1×	1.71×	3.42×	0.18×	1.15×
			class B	1×	1.73×	3.44×	0.18×	1.16×
			All	**1×**	**1.68×**	**3.33×**	**0.18×**	**1.12×**
Ultrafast configuration	Relative energy consumption		UVG	1×	0.36×	0.24×	0.89×	0.15×
			class A	1×	0.36×	0.22×	0.83×	0.13×
			class B	1×	0.35×	0.22×	0.85×	0.13×
			All	**1×**	**0.36×**	**0.23×**	**0.87×**	**0.14×**
	Encoding speed	fps	UVG	46	90	165	8	50
			class A	72	139	265	14	85
			class B	144	286	531	28	169
		speedup	UVG	1×	1.97×	3.60×	0.18×	1.10×
			class A	1×	1.94×	3.69×	0.20×	1.18×
			class B	1×	1.97×	3.68×	0.20×	1.18×
			All	**1×**	**1.96×**	**3.64×**	**0.19×**	**1.14×**

acceleration [23]. Table 3 shows that using one and two FPGAs with the Airframe platform increases encoding speed by 1.7× and 3.4×, respectively. With the Jetson platform, FPGA acceleration improves performance by a larger factor of 6.2×.

Combining the reduced power input with the increased encoding speed makes the proposed FPGA-accelerated HEVC encoder considerably more energy efficient, as analyzed next.

5 ENERGY SAVING RESULTS

Table 4 reports the results of energy consumption (kJ) and encoding speed (fps) for the QP values of 22 and 37. Additionally, Fig. 3 illustrates the aggregated energy used for encoding all test sequences according to QP value, platform, and encoding configuration. Table 5 reports the relative energy consumption (in relation to anchor), average frame rate, and speedup per dataset. In Table 5, the energy consumption was aggregated, and frame rate was averaged, across all QPs and sequences, per dataset. In Fig. 3 and Table 5, the SW-only configuration of the Airframe platform is used as the anchor for all configurations.

Based on the reported results, it can be concluded that using FPGA-accelerated HEVC encoder with both platforms reduces energy consumption and increases encoding speed regardless of the QP value and the dataset used. Using two FPGAs with the Airframe platform is the most beneficial setup. For example, with the *Fast* configuration and QP 22, the aggregated energy consumption of all test sequences for the SW-only configuration is 566.2 kJ, and it decreases to 218.3 kJ with one FPGA and 133.4 kJ with two FPGAs. On average, using one FPGA reduces the energy consumption by 0.39× over the SW-only configuration, and using two FPGAs reduces the energy consumption by an additional 0.61×, totaling to a reduction of 0.24×. The aggregated energy consumption of the SW-only configuration for all test sequences on the Jetson platform with *Fast* configuration and QP 22 is 534.3 kJ, which is only a 0.94× reduction over the Airframe platform. This is likely due to Kvazaar being AVX2-optimized and does not thereby benefit from the ARM architecture, resulting in longer encoding times. However, deploying FPGA reduces the energy consumption of the Jetson platform to 82.3 kJ, making it the most energy-efficient platform configuration in our experiments. The lack of AVX2-optimizations in this case does not hinder the Jetson platform, as the HEVC intra encoder is running on FPGA.

The FPGA-accelerated HEVC encoder not only reduces energy consumption, but also significantly improves encoding speed on both the Airframe and Jetson platforms. According to Table 5, the maximum average speedups for both platforms with the *Fast* configuration are 3.33× and 6.22×, respectively, compared with the SW-only solution of the same platform. In addition, Table 5 indicates that on average, the Airframe and Jetson platforms can achieve respective frame rates of up to 165 fps and 50 fps for UVG 4K sequences in *Ultrafast* configuration.

6 CONCLUSION

This paper analyzed the energy efficiency of an FPGA-accelerated HEVC intra encoder on two different platforms: Nokia Airframe edge server with dual 14-core Xeon CPU and Jetson AGX Orin edge device with 12-core ARM processor. Our experimental results showed that deploying dedicated HW can significantly reduce energy consumption on both platforms, and also speed up the encoding process. With *Fast* encoding configuration, we achieved overall energy reduction of up to 76% (0.24×) with Airframe using two FPGAs and 86% (0.14×) with Jetson using one FPGA compared with CPU only encoding on Airframe. The respective 4K coding speed improved by 3.3× and 6.2×. For future work, we plan to customize the HW encoder for *Region-Of-Interest* (*ROI*) coding, which we believe will further decrease energy consumption and improve coding speed for the same visual quality.

REFERENCES

[1] ETSI, "Multi-access Edge Computing (MEC)," [Online]. Available: https://www.etsi.org/technologies/multi-access-edge-computing.

[2] X. Jiang, F. R. Yu, T. Song, and V. C. M. Leung, "A survey on multi-access edge computing applied to video streaming: some research issues and challenges," *Commun. Surveys Tuts.*, vol. 23, no. 2, Mar. 2021, pp. 871–903.

[3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 13, no. 7, Aug. 2003, pp. 560–576.

[4] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 22, no. 12, Dec. 2012, pp. 1649–1668.

[5] ITU, "New 'Versatile Video Coding' standard to enable next-generation video compression," Sept. 2020, [Online]. Available: https://www.itu.int/en/mediacentre/Pages/pr13-2020-New-Versatile-Video-coding-standard-video-compression.aspx.

[6] Bitmovin, "Bitmovin video developer report 2019," 2019, [Online]. Available: https://cdn2.hubspot.net/hubfs/3411032/Bitmovin%20Magazine/Video%20Developer%20Report%202019/bitmovin-video-developer-report-2019.pdf.

[7] F. Saab, I. H. Elhajj, A. Kayssi, and A. Chehab, "Energy analysis of HEVC compression with stochastic processing," in *Proc. IEEE Mediterranean Electrotechnical Conf.*, Beirut, Lebanon, Apr. 2014.

[8] E. Monteiro, M. Grellert, S. Bampi, and B. Zatt, "Rate-distortion and energy performance of HEVC and H.264/AVC encoders: a comparative analysis," *in Proc. IEEE Int. Symp. Circuits Syst.*, Lisbon, Portugal, May 2015.

[9] M. Uitto, "Energy consumption evaluation of H.264 and HEVC video encoders in high-resolution live streaming," in *Proc. IEEE Int. Conf. Wireless Mobile Comput.*, *Netw. Commun.*, New York, New York, USA, Oct. 2016.

[10] A. Mercat, F. Arrestier, W. Hamidouche, M. Pelcat, and D. Menard, "Energy reduction opportunities in an HEVC real-time encoder," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process.*, New Orleans, Louisiana, USA, Mar. 2017.

[11] D. Silveira, M. Porto, and S. Bampi, "Performance and energy consumption analysis of the x265 video encoder," in *Proc. European Signal Process. Conf.*, Kos, Greece, Aug. 2017.

[12] R. Mathá, D. Kimovski, A. Zabrovskiy, C. Timmerer, and R. Prodan, "Where to encode: a performance analysis of x86 and arm-based amazon EC2 instances," in *Proc. IEEE Int. Conf. eScience*, Innsbruck, Austria, Sept. 2021.

[13] R. Rodríguez-Sánchez, M. T. Alonso, J. L. Martínez, R. Mayo, and E. S. Quintana-Ortí, "Time and energy modeling of an intra-only HEVC encoder," in *Proc. IEEE Visual Commun. Image Process.*, Singapore, Singapore, Dec. 2015.

[14] G. Ramasubbu, A. Kaup, and C. Herglotz, "Modeling the HEVC encoding energy using the encoder processing time," in *Proc. IEEE Int. Conf. Image Process.,* Bordeaux, France, Oct. 2022.

[15] G. Ramasubbu, A. Kaup, and C. Herglotz, "A bit stream feature-based energy estimator for HEVC software encoding," in *Proc. Picture Coding Symp.*, San Jose, California, USA, Dec. 2022.

[16] E. Kalali, Y. Adibelli, and I. Hamzaoglu, "A high performance and low energy intra prediction hardware for high efficiency video coding," in *Proc. Int Conf. Field Programmable Logic Appl.*, Oslo, Norway, Aug. 2012.

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

[18] 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, Nov. 2014, pp. 745–753.

[19] 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, Nov. 2014, pp. 754–761.

[20] 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, May 2016, pp. 166–174.

[21] I. Seidel, A. B. Bräscher, J. L. Güntzel, and L. Agostini, "Energy-efficient SATD for beyond HEVC," in *Proc. IEEE Int. Symp. Circ. Syst.*, Montreal, Quebec, Canada, May 2016.

[22] S. S. Deepsita, K. Divya, and S. N. Mahammad, "Energy efficient and multiplierless approximate integer DCT implementation for HEVC," in *Proc. IEEE Int. Conf. Very Large Scale Integr.*, Singapore, Singapore, Oct. 2021.

[23] P. Sjövall, A. Lemmetti, J. Vanne, S. Lahti, and T. D. Hämäläinen, "High-level synthesis implementation of an embedded real-time HEVC intra encoder on FPGA for media applications," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 27, no. 35, Mar. 2022.

[24] A. Lemmetti, M. Viitanen, A. Mercat, and J. Vanne, "Kvazaar 2.0: fast and efficient open-source HEVC inter encoder," in *Proc. ACM Multimedia Syst. Conf.*, Istanbul, Turkey, May 2020.

[25] Ultra Video Group, "Kvazaar open-source HEVC encoder," [Online]. Available: https://github.com/ultravideo/kvazaar.

[26] P. Coussy, D. D. Gajski, M. Meredith, and A. Takach, "An introduction to high-level synthesis," *IEEE Des. Test. Comput.*, vol. 26, no. 4, Jul. 2009, pp. 8–17.

[27] Siemens, "Catapult high-level verification solutions," [Online]. Available: https://eda.sw.siemens.com/en-US/ic/catapult-high-level-synthesis/hls-verification/

[28] F. Bossen, J. Boyce, K. Suehring, X. Li, and V. Seregin, "JVET common test conditions and software reference configurations for SDR video," *document JVET-N1010*, Geneva, Switzerland, Mar. 2019.

[29] C. C. Chi *et al*., "Parallel scalability and efficiency of HEVC parallelization approaches," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, Dec. 2012, pp. 1827–1838.

[30] A. Mercat, M. Viitanen, and J. Vanne, "UVG dataset: 50/120fps 4K sequences for video codec analysis and development," in *Proc. ACM Multimedia Syst. Conf.*, Istanbul, Turkey, May 2020.

[31] G. Bjøntegaard, "Calculation of average PSNR differences between RD curves," *Document VCEG-M33*, Austin, Texas, USA, Apr. 2001.

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