

High-Throughput Multifilter Interpolation Architecture for AV1 Motion Compensation

Robson Domanski, Jones Goebel, Wagner Penny, *Student Member, IEEE*, Marcelo Porto, *Senior Member, IEEE*, Daniel Palomino, *Member, IEEE*, Bruno Zatt, *Senior Member, IEEE*, and Luciano Agostini, *Senior Member, IEEE*,

Abstract—This paper presents a high-throughput hardware design for the Motion Compensation (MC) sub-sample interpolator of the novel AOMedia Video 1 (AV1). A Multifilter Interpolation Architecture (MFIA) is presented targeting AV1 decoders and supporting all 90 filters defined by the AV1 MC. The MFIA uses shift-adds instead of multipliers and common subexpression sharing to reduce the hardware costs, to increase the throughput and to reduce the power dissipation. Two different versions of MC interpolators using MFIA are also presented in this paper. The MFIA and the two versions of MC interpolators were synthesized for 40nm TSMC standard-cells. The results showed that the MC interpolators are able to process up to 4320p videos at 30fps, with a maximum power dissipation of 81.31mW. To the best of the authors' knowledge, this is the first work presenting a hardware design for the MC of AV1.

Index Terms—hardware design, interpolation, motion compensation, video coding, AV1

I. INTRODUCTION

OVER the last years, the number of mobile devices capable of capturing, manipulating, storing, and transmitting digital videos has grown expressively. Nowadays, more than a half of video visualizations happen at mobile devices [1], which made them responsible for Internet traffic that exceeded 7 EiB/month in 2016, with an expectation of reaching up to 49 EiB/month in 2021 (where 1EiB is equal to 2^{60} bytes) [2]. Indeed, according to Cisco [3], in 2021, 80-90% of all global Internet traffic will be related to video sharing. Driven by such expressive growth on video demand, intense research effort was employed to increase the video coding efficiency, focusing on videos with high resolutions and high frame rates, leading to the specification of new video coding standards.

The High Efficiency Video Coding (HEVC) [4] standard is the ITU-T and ISO/IEC state-of-the-art video encoder and it was released in 2013. The HEVC reaches around 40% higher

compression rates for the same video quality in relation to ITU-T and ISO/IEC previous standard, the H.264/AVC [5] [6]. However, at the same time that ITU-T and ISO/IEC experts were working to improve the previous standards, some companies started to develop their own encoders, mainly intending to have royalty-free video codecs. Google was especially involved in this type of effort, and in 2013, the same year of HEVC release, the VP9 [7] was released by Google, as a new generation of Google encoders.

The HEVC contains several patented technologies from a range of individuals and/or parties whose licensors are also different, leading to the undesirable payment of royalties for its usage [8]. Due to this fact, several giant companies like Google, Intel, Cisco, Mozilla, Microsoft, Netflix, Amazon, and many others, have formed the Alliance for Open Media (AOM) aiming the development of a new royalty-free video codec called AOMedia Video 1 (AV1) [9]. The AV1 included innovations of VP10, which was being developed as the VP9 successor, and innovations from other pattern companies, like Mozilla's Daala [10] and Cisco's Thor [11] encoders. AV1 first bitstream and decoding specification was released in April 2018, establishing the reference software AOMedia Codec version 1.0.0 [12]. This open-source video codec was conceived to be completely royalty-free and scalable to any modern device at any bandwidth. AV1 was also expected to be flexible for use in both commercial and non-commercial content, optimal for video broadcasting and hardware designs, and consistent for Ultra High Definition (UHD) resolutions, reaching a compression efficiency up to 30% higher than VP9 and HEVC encoders [13][14].

Regarding current video codecs, such as HEVC, VP9, and AV1, one can notice that all of them define complex coding tools, which hinders software implementations, mainly when high-resolution videos must be processed in real-time, and/or the codecs are running on battery-powered devices. Thus, dedicated hardware designs are mandatory to deal with such complex tasks. Considering such scenario, the Motion Compensation (MC) can be detached as one of the most time-consuming step within the decoder side. MC is used at both encoder and decoder sides to reconstruct the frames considering the information generated by the Motion Estimation (ME). Since current video encoders support Fractional Motion Estimation (FME), the MC must be able to interpolate the integer samples to generate the required fractional samples. This interpolation is done through Finite Impulse Response (FIR) filters, which are the main responsible for the computational

R. Domanski, J. Goebel, M. Porto, D. Palomino, B. Zatt and L. Agostini are with the Video Technology Research Group (ViTech), Federal University of Pelotas (UFPEL), Rua Gomes Carneiro 1, Pelotas-RS, 96010-610, Brazil (e-mails: {radomanski, jwgoebel, porto, dpalomino, zatt, agostini}@inf.ufpel.edu.br).

W. Penny is with the Sul-rio-grandense Federal Institute (IFSul), Praça Vinte de Setembro 455, Pelotas-RS, 96015-360, Brazil; and with the Video Technology Research Group (ViTech), Federal University of Pelotas (UFPEL), Rua Gomes Carneiro 1, Pelotas-RS, 96010-610, Brazil (e-mail: wi.penny@inf.ufpel.edu.br).

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance code 001, the Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil (CNPq), and the Fundação de Amparo à Pesquisa do Rio Grande do Sul - Brasil (FAPERGS).

effort required in the MC processing. For instance, the MC is responsible for 38% of the HEVC computational effort [15] and more than 50% in the H.264/AVC [16].

Many works in the literature [17], [18], [19], [20] aim the development of hardware designs targeting the motion compensation in video coding. Works [17], [18], [19] developed ASIC-based hardware designs for the HEVC MC sample interpolator. Work [20] also designed an MC sample interpolator hardware but targeting Google VP9-10 video codecs. Therefore, to the best of our knowledge, there is no work in the literature that presents a hardware design targeting the AV1 MC sample interpolator.

This work presents a multifilter hardware design that can efficiently process all available AV1 MC filtering operations, supporting all 90 filters applied over luminance (luma) and chrominance (chroma) information. This solution was named as Multifilter Interpolation Architecture (MFIA) in this work. Varying the number of MFIA instances in an AV1 MC interpolation design will bring different relations among hardware costs, power dissipation and processing rates. Two different complete AV1 MC interpolation designs are also presented in this work, targeting different processing rates of 2160p@30fps and 4320p@30fps.

II. AV1 AND MOTION COMPENSATION BACKGROUND

AV1 divides the frame into blocks of different sizes (square and rectangular shaped), ranging from 4x4 up to 128x128 [9][12]. These blocks are called *SuperBlocks* (SBs) and are the basic AV1 coding units. The SBs are the root of the partitioning tree in the coding process and the quadratic blocks can be recursively partitioned till the 4x4 size.

The AV1 inter prediction contains the Motion Estimation (ME) and the Motion Compensation (MC) steps. The ME is for finding good predictors for integer (IME) and fractional (FME) samples and defining the Motion Vectors (MV). The IME uses the samples available in the reference frames to define the best matching when compared with the current block samples. The FME uses fractional samples generated around the IME best block to do a second matching step to define the final best block. The fractional samples are generated by an interpolation process, which is performed by the application of Finite Impulse Response (FIR) filters over the integer samples of the reference frame. At the end of IME and FME of a current block, a Motion Vector (MV) is defined pointing to the final best block of the ME process.

The MC has the responsibility to reconstruct the blocks predicted by the ME, using the MVs as main input information. The reconstructed blocks are necessary in both encoder and decoder sides. At the encoder side the reconstructed blocks are necessary to be used as reference for the ME of next frames. At the decoder side, besides of being used as reference, these are also the blocks that will be shown to the users. MC calculates the block samples (both luminance and chrominance samples) using the selected MVs and the reference frames.

The interpolation process used in AV1 MC is based on FIR and bi-linear filters. AV1 defines filters with up to 8-taps, achieving an accuracy of 1/8 and 1/16 samples, for luminance

and chrominance, respectively [12]. Such accuracies are considerably higher when compared to other current video coding standards, such as HEVC or H.264, which have an accuracy of 1/4 (luminance) and 1/8 (chrominance) samples [4][5].

The AV1 also presents a larger number of MC filter families [12], when compared with other codecs. These sets of families are: (i) **Regular**: a 6-tap filter which consists of a Lagrange-based interpolation filter; (ii) **Smooth**: a 6-tap filter which consist of a smoothing filter designed in Hamming Window; (iii) **Sharp**: an 8-tap filter which consists of an interpolation filter based on a Direct Cosine Transform (DCT); and (iv) **Bi-linear**: which is used when fast encoding or decoding is required.

Furthermore, AV1 defines a simplification of the Regular, Smooth and Sharp filters families when the block size dimension (width and/or height) of the interpolated block is equal or less than four. The main simplification in these cases is the reduction of used filter taps, allowing the use of only 4-taps.

Therefore, the interpolation process in the AV1 can select six sets of filters (considering four families), which totalize 90 filters [12]. It is important to notice that AV1 uses the *same* filters for luminance and chrominance samples, the only difference is the precision used for each sample. Moreover, it is important to highlight that coefficients used in the calculation of 5/8, 6/8, and 7/8 luminance samples are similar to the ones used in the calculation of 1/8, 2/8, and 3/8, but rotating the coefficients (the filters are symmetric). Similarly, the coefficients used in the calculation of 9/16 to 15/16 chrominance samples are the same used in 1/16 to 7/16 but rotating the coefficients.

The high precision defined by AV1 together with the high number of available filters lead to interesting improvements on the compression efficiency, however, it also increases the required computational effort, which makes the processing of high-resolution videos, regarding real-time scenarios, a great challenge.

III. AV1 MC MULTIFILTER ARCHITECTURE

The main contribution of this work is the development of an optimized AV1 MC Multifilter Interpolation Architecture (MFIA), which efficiently supports all filtering calculations required by the AV1 MC decoder. This means that MFIA supports all 90 filters defined by AV1 MC and can process luminance and chrominance samples. The MFIA was also designed thinking in different performance targets, since the use of multiple instances of MFIA allow different targets of processing rate, as will be discussed in the next section.

The MFIA architecture, presented in Fig. 1, firstly explores the conversion of all multiplicative coefficients defined by the AV1 MC in shift-adds. Then, the common subexpressions are shared among the different filters. Both solutions contribute to reduce hardware use, to reduce power consumption and to increase the architecture throughput.

As explained before, the AV1 MC filters are symmetric and such behavior was also explored into the MFIA design. This symmetry means that, for example, the filter at position 1/16 uses the same coefficients than the filter at position 15/16, only changing the input sequence. For this reason, it is possible to

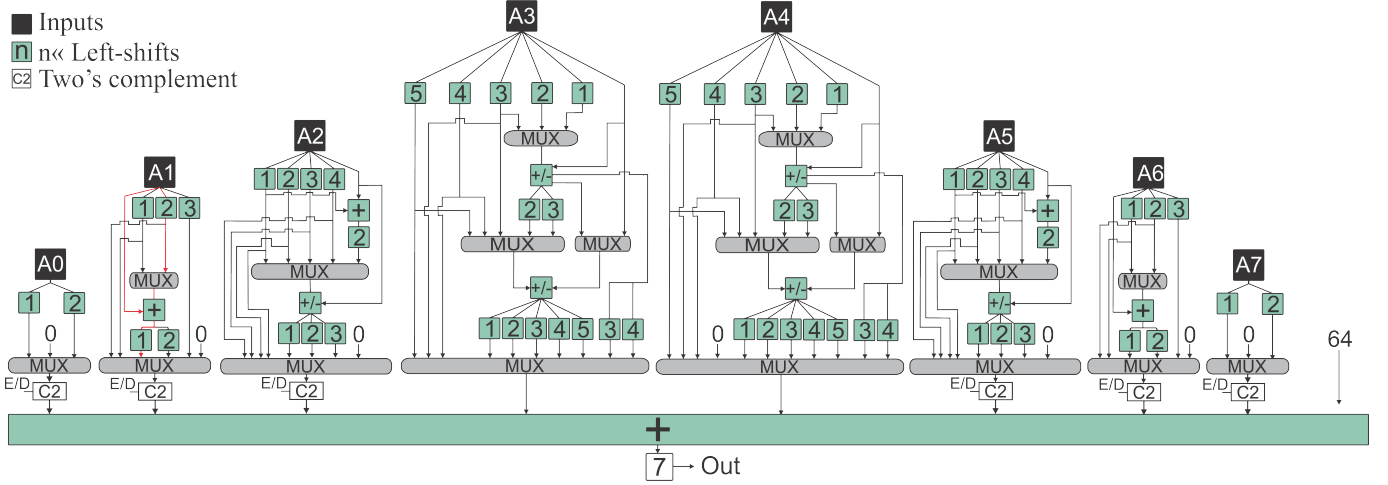


Fig. 1. Proposed AV1 MC Multifilter Interpolation Architecture.

share the same hardware for both filters through the selection of the correct inputs.

The selection of which calculation must be done is defined based on MV_x and MV_y input values, which are the x and y components of the MV for each block. MV_x is used to define the horizontal filtering and MV_y is used to define the vertical filtering. The MV_x and MV_y can range from 0 to 15 positions to support all possible fractional samples positions. The MFIA module has also one additional input control that is responsible to define which filter family must be performed, thus running a filter for each block. This information is available in the bitstream of the encoded video for each block size.

In Fig. 1, A0-A7 represent the input samples of the filter and the Out represents the processed sample output. Each input-sample value can be multiplied by different constants, according to the type of filter that is configured to compose the output sample. For instance, the sample A1 can be multiplied by: (i) the constant 2 (for 6-tap Regular, 6-tap Smooth and 8-tap Sharp filters); (ii) by constants 4, 6, 8, 10 and 12 (for 8-tap Sharp filters); or (iii) constant 0 for other filters (considering also filters with 4-taps). Then, the multiplexers are responsible to select which operation result (shift-add) will be performed according the filter type. As an example, (highlighted in red wires in Fig. 1) the input A1 is multiplied by 10 with the operations $(A1 + (A1 << 2)) << 1$. The operations required for sample A1 to be multiplied by 10 are: (i) multiplication by 2: $A1 << 1$; (ii) multiplication by 4: $A1 << 2$; (iii) multiplication by 6: $(A1 + (A1 << 1)) << 1$; (iv) multiplication by 8: $A1 << 3$; (v) multiplication by 12: $(A1 + (A1 << 1)) << 2$.

Fig. 1 also shows that the multiplexer output is connected to a 2-complement operation because subtraction may be required in some filters. The last adder performs the sum of all samples and delivers the result according to the selected filter. Finally, a 7-bits shift right is performed as a clip operation. As one can notice from Fig. 1, the MFIA was designed in a purely combinational fashion, allowing the processing of eight input samples in parallel.

IV. AV1 MC INTERPOLATOR ARCHITECTURES

This section describes of design two architectures for the AV1 decoder MC interpolator using the MFIA solution. Those architectures must support all 90 filter types but at the decoder side just one filter must be applied to each block size, according to the encoder decision informed at the encoded bitstream. Furthermore, both architectures were designed to process 4x4 block sizes, which is the smallest block size supported by AV1. By using such block size, it is possible to support all available AV1 block sizes by splitting larger blocks into 4x4 blocks. The difference between the two design is the level of parallelism supported in each one of them. The first solution is called 1RC (one row per cycle) and it processes one row at each clock cycle, using eight instances of MFIA. The second solution processes four rows per cycle and is called 4RC (four row per cycle), requiring 32 instances of MFIA. Note that those designs are completely independent, reaching different goals.

Fig. 2 illustrates the architecture of the first design of the top-level interpolator denominated 1RC, which is composed of four horizontal MFIA filters ($MFIA_H$ in Fig. 2), and four vertical MFIA filters ($MFIA_V$ in 2), and four shift-register chain (SRC) to connect the filters. Each black square in the Fig. 2 represents the integer input (luma or chroma) samples. $MFIA_H$ and $MFIA_V$ are identical, using the architecture presented in Fig. 1. The different names are used just to highlight which instances are used in the vertical and horizontal filtering processes.

The architecture operates with 4x4 block sizes, requiring an input matrix of 11x11 samples in the worst case (generally, filters with n taps requires $n-1$ sample per border besides the block being processed). The 1RC reads one row per cycle (eleven samples) from the input matrix and those samples are used to feed the input of the $MFIA_H$ instances. Each $MFIA_H$ instance receives eight input samples and generates one output sample. The output sample of each $MFIA_H$ instance is connected to a shift-register chain, which is composed by eight registers, as shown in Fig. 2. The shift-register chain is responsible to feed the correct value/position to each $MFIA_V$

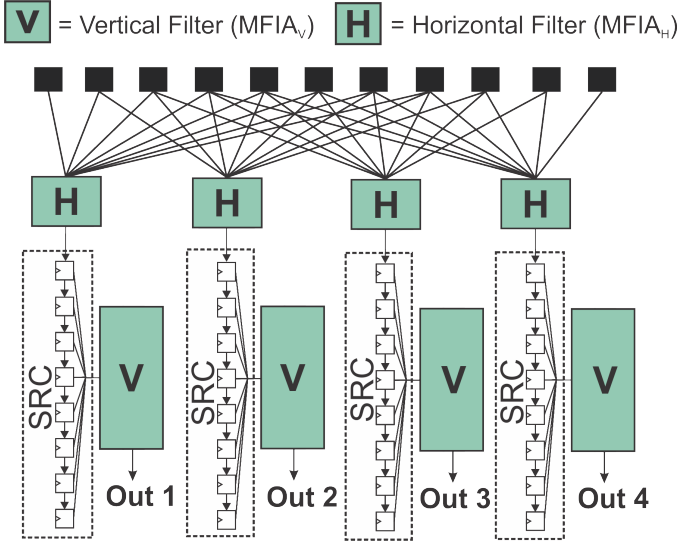


Fig. 2. Block diagram of 1RC architecture with four MFIA instances.

instance. Finally, the $MFIA_V$ outputs are the MC interpolator outputs.

The 1RC architecture has a latency of seven cycles and then, between cycles eight to eleven, it delivers four valid samples per cycle. Then, eleven cycles are necessary to deliver the interpolation of a 4×4 input block.

Fig. 3 shows the second top-level architecture of the developed interpolator, termed 4RC, which is designed to have a level of parallelism greater than 1RC, being capable of processing four lines per cycle. The 4RC was designed with 16 $MFIA_H$ instances, 16 $MFIA_V$ instances, and four shift-register chain routers (SRCR in Fig. 3). This architecture was also designed to process 4×4 block sizes and receive 44 samples per cycle from the input matrix, which is composed of 11×11 samples. The 44 input samples are presented as gray-scale squares in Fig. 3.

Each SRCR presented in Fig. 3 is responsible to store the samples that come from four $MFIA_H$ instances, routing those samples to the $MFIA_V$ instances. To perform this processing,

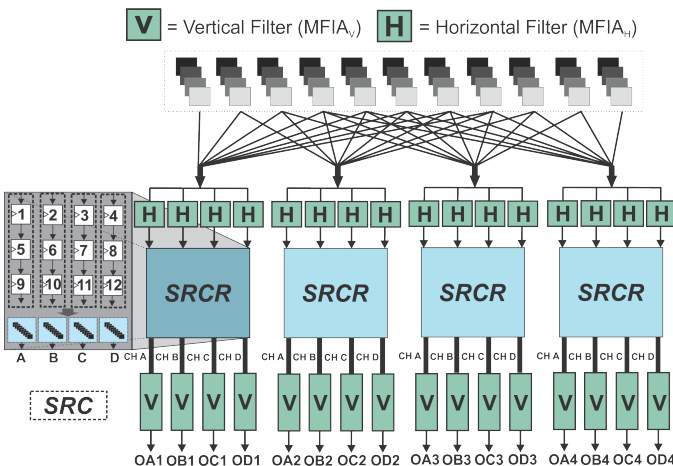


Fig. 3. Block diagram of 4RC architecture with four 32 MFIA instances.

the SRCR has four shift-register chains (SRC) with three registers each, and each SRC is responsible to store the correspondent output sample from $MFIA_H$ instance. The first $MFIA_H$ instance is connected as input of register 1, the second $MFIA_H$ instance is connected as input of register 2, and so on. In the same way, the first $MFIA_H$ instance is responsible to process the first row of input samples, the second $MFIA_H$ instance processes the second row of input samples, up to reach the four rows. With all samples stored in the SRCR, it is possible to route those samples to the $MFIA_V$ instances. The input of the first $MFIA_V$ instance is connected to registers 12, 11, 10, 9, 8, 7, 6 and 5. The input of second $MFIA_V$ instance is connected to registers 11, 10, 9, 8, 7, 6, 5 and 4. The third $MFIA_V$ instance is connected to registers 10, 9, 8, 7, 6, 5, 4 and 3. Finally, the fourth $MFIA_V$ module is connected to registers 9, 8, 7, 6, 5, 4 and 2. One can notice the register 1 is not used in the routing to $MFIA_V$ instances and such additional register is required to adjust the level of parallelism between the $MFIA_H$ and $MFIA_V$.

The outputs of the 4RC are composed of the outputs delivered by all $MFIA_V$ instances. This architecture has a latency of two cycles and in the third cycle it delivers the 16 valid outputs. In comparison to the first solution, the 4RC needs only three cycles to process a 4×4 block against eleven cycles.

V. RESULTS AND COMPARISONS WITH RELATED WORKS

The MFIA, the 1RC and 2RC complete interpolator architectures were described in VHDL and were synthesized for a 40nm TSMC standard-cells technology with 1.1V [21] using Cadence RTL Compiler tool [22]. The power results were generated using the default tool switching activity (20%). The gate count was calculated based on 2-input NANDs that represent $0.9408 \mu m^2$ for TSMC.

The synthesis results are shown in Table I. The MFIA results showed an area usage of 4.96Kgates and a power dissipation of 2.45mW when running at 279.9MHz (our highest target frequency).

The 1RC interpolator was designed to process 4K (3840x2160 pixels) UHD videos at 30fps. As was explained before, such architecture needs eleven cycles to generate one 4×4 block, requiring a minimum operational frequency of 256.6 MHz to reach the target performance. On the other hand, the 4RC needs three cycles to generate one 4×4 block and it was designed to process 8K UHD videos at 30fps, then it must run at 279.9 MHz to reach such target.

TABLE I
SYNTHESIS RESULTS

Architecture	Freq. (MHz)	Gates (K)	Power (mW)	Throughput
MFIA	279.9	4.96	2.45	-
1RC	256.6	40.70	16.12	2160@30fps
4RC	279.9	141.10	81.31	4320@30fps

When comparing the 4RC and 1RC, one can notice that 4RC uses almost 3.5 times more area and dissipates five times more power than 1RC. However, 4RC can process 8K UHD videos at 30fps, which means that 4RC can process 4 times more samples than 1RC with a higher throughput.

In addition, there is no other work published in the literature that presents a hardware design for AV1 MC sample interpolator, and then, it is not possible to provide a fair comparison of our work with related works. A comparison with works targeting the H.264, HEVC and VP9-VP10 MC interpolations is not easy, since the AV1 MC interpolation is much more complex.

The work [15] presents an interpolator for the HEVC MC supporting eleven filters. When synthesized for a 65nm TSMC technology it uses 98.68K gates and requires a power of 29.12mW to process 4320p videos at 60fps. Since our solution must support much more filters, it is expected to be more area and power demanding.

The work [19] is a multi-standard solution, supporting the HEVC and the H.264/AVC MC interpolations. Only luminance samples are supported, and the design includes four FIR and two bi-linear filters. The synthesis for 65nm TSMC technology uses 166.8K gates with a power consumption of 80.69mW to process 2160p videos at 60fps. Even supporting much more filters, our 4RC interpolator has similar results for area, power and processing rates.

Finally, the work [20] supports 45 FIR and 15 bilinear filters, which is still a lower number of filters than supported by our solution (90 filters). This work was synthesized for 45nm Nangate technology and uses 71.2 K gates, consuming 2.34mW to process 2160p videos at 60fps. The power estimation used real input samples and considered the use of low-power techniques.

VI. CONCLUSION

This paper presented the Multifilter Interpolation Architecture for the AV1 motion compensation decoder. The MFIA supports all 90 filters defined by the AV1 MC and uses shift-adders and common subexpression sharing to reduce the hardware costs, to reduce the power dissipation and to increase the throughput. This paper also presented two versions of complete MC interpolation architectures targeting different processing rates using different number of MFIA instances.

The MFIA and the complete MC interpolation architectures were synthesized targeting the 40nm TSMC standard-cell technology and the results showed that the MC interpolators are able to process 2160p and 4320p videos at 30fps with a power dissipation of 16.12mW and 81.31mW, respectively. The reached results were competitive with related works, even considering that our work supports an interpolation process with a higher level of complexity when compared with previous standards related works.

ACKNOWLEDGMENT

This work was supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and also by FAPERGS and CNPq Brazilian research support agencies.

REFERENCES

- [1] Youtube. (2018, oct) Youtube statistics. [Online]. Available: <https://www.youtube.com/intl/pt-BR/yt/about/press/>.
- [2] Statista. (2018, oct) Mobile internet: Statistics and facts on mobile internet usage, 2016, 2017. [Online]. Available: <https://www.statista.com/topics/779/mobile-internet/>.
- [3] Cisco. (2018, oct) Cisco visual networking index: Forecast and methodology, 2016–2021 - cisco. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html>.
- [4] ISO/IEC-JCT1/SC29/WG11, *High Efficiency Video Coding (HEVC) text specification draft 10.*, doc jctvc-I1003. ed., 2013.
- [5] 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 *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2015, pp. 1278–1281.
- [6] D. Grois, D. Marpe, A. Mulayoff, B. Itzhaky, and O. Hadar, “Performance comparison of h.265/mpeg-hevc, vp9, and h.264/mpeg-avc encoders,” in *2013 Picture Coding Symposium (PCS)*, Dec 2013, pp. 394–397.
- [7] D. Mukherjee, J. Han, J. Bankoski, R. Bultje, A. Grange, J. Koleszar, P. Wilkins, and Y. Xu, “A technical overview of vp9 – the latest open-source video codec,” in *SMPTE 2013 Annual Technical Conference Exhibition*, Oct 2013, pp. 1–17.
- [8] M. A. Layek, N. Q. Thai, M. A. Hossain, N. T. Thu, L. P. Tuyen, A. Talukder, T. Chung, and E. Huh, “Performance analysis of h.264, h.265, vp9 and av1 video encoders,” in *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, Sep. 2017, pp. 322–325.
- [9] AOMedia. (2018, oct) Alliance for open media. [Online]. Available: <https://aomedia.org/>.
- [10] XIPH.ORG. (2018, oct) Daala video compression. [Online]. Available: <https://xiph.org/daala>.
- [11] G. Bjøntegaard, T. Davies, A. Fuldseth, and S. Midtskogen, “The thor video codec,” in *2016 Data Compression Conference (DCC)*, March 2016, pp. 476–485.
- [12] P. Rivaz and J. Haughton, *AV1 Bitstream & Decoding Process Specification. Version: 1.0.0. Copyright 2018, The Alliance For Open Media*, 2018, 677 p.
- [13] AOMedia. (2018, oct) Av1 features. [Online]. Available: <https://aomedia.org/av1-features/>.
- [14] A. Zbrovskiy, C. Feldmann, and C. Timmerer, “Multi-codec dash dataset,” in *Proceedings of the 9th ACM Multimedia Systems Conference*, ser. MMSys '18. New York, NY, USA: ACM, 2018, pp. 438–443. [Online]. Available: <http://doi.acm.org/10.1145/3204949.3208140>
- [15] C. M. Diniz, M. Shafique, S. Bampi, and J. Henkel, “A reconfigurable hardware architecture for fractional pixel interpolation in high efficiency video coding,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 2, pp. 238–251, Feb 2015.
- [16] E. Li and Y.-K. Chen, “Implementation of h.264 encoder on general-purpose processors with hyper-threading technology,” vol. 5308, 01 2004, pp. 384–395.
- [17] 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 Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 2, pp. 380–393, Feb 2017.
- [18] W. Penny, G. Paim, M. Porto, L. Agostini, and B. Zatt, “Real-time architecture for hevc motion compensation sample interpolator for uhd videos,” in *2015 28th Symposium on Integrated Circuits and Systems Design (SBCCI)*, Aug 2015, pp. 1–6.
- [19] G. Pastuszak and M. Trochimiuk, “Architecture design of the high-throughput compensator and interpolator for the h.265/hevc encoder,” *Journal of Real-Time Image Processing*, vol. 11, no. 4, pp. 663–673, Apr 2016. [Online]. Available: <https://doi.org/10.1007/s11554-014-0422-1>
- [20] G. Paim, W. Penny, J. Goebel, V. Afonso, A. Susin, M. Porto, B. Zatt, and L. Agostini, “An efficient sub-sample interpolator hardware for vp9-10 standards,” in *2016 IEEE International Conference on Image Processing (ICIP)*, Sep. 2016, pp. 2167–2171.
- [21] TSMC. (2018, aug) 40nm technology. [Online]. Available: <http://www.tsmc.com/english/dedicatedFoundry/technology/40nm.htm>.
- [22] Cadence. (2018, aug) Encounter rtl compiler. [Online]. Available: <http://www.cadence.com>.