

Modern Video Coding: Methods, Challenges and Systems

Roberta Palau, Bianca Silveira, Robson Domanski, Marta Loose, Arthur Cerveira, Felipe Sampaio, Daniel Palomino, Marcelo Porto, Guilherme Corrêa, Luciano Agostini

Group of Architectures and Integrated Circuits – GACI, Video Technology Research Group – ViTech
Graduate Program in Computing – PPGC, Federal University of Pelotas – UFPEL, Brazil
e-mail: rcnpalau@inf.ufpel.edu.br

Abstract— With the increasing demand for digital video applications in our daily lives, video coding and decoding become critical tasks that must be supported by several types of devices and systems. This paper presents a discussion of the main challenges to design dedicated hardware architectures based on modern hybrid video coding formats, such as the High Efficiency Video Coding (HEVC), the AOMedia Video 1 (AV1) and the Versatile Video Coding (VVC). The paper discusses each step of the hybrid video coding process, highlighting the main challenges for each codec and discussing the main hardware solutions published in the literature. The discussions presented in the paper show that there are still many challenges to be overcome and open research opportunities, especially for the AV1 and VVC codecs. Most of these challenges are related to the high throughput required for processing high and ultra-high resolution videos in real time and to energy constraints of multimedia-capable devices.

Index Terms— Video Coding, Hardware Design, HEVC, AV1, VVC.

I. INTRODUCTION

Nowadays, video coding is essential, since digital videos are currently spread in many professional and entertainment applications, with high video resolution and frame rate requirements, which are prohibitive without the use of compression techniques. The current projections point out that video data will represent 90% of all internet traffic by 2023 [1]. But this forecast can be even worse: since social isolation due to the covid-19 pandemic started, video consumption and internet traffic have increased a lot. For example, Youtube and Netflix needed to reduce video quality to guarantee the quality of service [2]. This shows the increasing importance of video coding nowadays.

To support the continuous increase in video resolution and frame rate, many codecs have been developed in the last years to reach high video coding efficiency (i.e., the tradeoff between video quality and bitrate), but at the cost of a significant increase in computational effort. To cope with this side effect, novel solutions are required to spread the use of these encoders and decoders, using graphic processor units (GPUs), image signal processors (ISPs) [3] and/or dedicated hardware accelerators. When battery-powered devices are targeted, the use of dedicated hardware is mandatory due to energy constraints. Either way, video content coding or decoding requires more than an efficient software running on a powerful general-purpose processor (GPP).

Hardware support for video coding is mainly present in systems-on-a-chip (SoC) targeting smartphones, but also in modern GPU and GPPs, since even the powerful GPUs or GPPs need dedicated acceleration to allow the coding and

decoding of digital videos in real time. Fig. 1(a) shows the die of the NVidia Tegra 2 SoC [4], launched in 2011. This die indicates where the video encoder and decoder are located. From this figure one can observe that even older SoCs already had support to dedicated hardware for video coding and decoding. This SoC supports two video codecs, the H.264/AVC and the Microsoft VC-1. Fig. 1(b) shows the Qualcomm Snapdragon 820 die [5], launched in 2016. The Spectra Image Signal Processor (ISP) comprises dedicated hardware that supports the HEVC and H.264/AVC standards [6]. Fig. 1(c) presents the Intel i7-1065G7 GPP die [7]. This processor was launched in 2020 and, as other previous Intel processors, uses the Quick Sync Video technology (inside the block “Imaging” in Fig. 1 (c)), which has hardware support for a variety of encoders, like H.264/AVC, HEVC, VP8 and VP9 [8]. Finally, some GPUs also have dedicated hardware acceleration for video coding, like the GeForce RTX 30 Series, launched in 2020, which is presented in Fig. 1(d) [9]. This GPU supports many video codecs, like MPEG-2, VC-1, H.264/AVC, HEVC, VP8, VP9 and AV1 [10].

Unfortunately, these commercial solutions do not present how the encoders and decoders are supported in hardware. Considering the extremely high complexity of current video codecs, one can imagine that a lot of simplifications are done, mainly at the encoder side, to allow the real-time processing of high-resolution videos and to reach an energy consumption as lower as possible. On the other hand, some academic works detail the proposed solutions, but most of them do not support all the encoder or decoder tools. Most of them are only focused on one of these tools, like the works [11], [12], [13], [14], which employ low-power techniques, and [15], [16], [13], which use approximate computing.

Considering this scenario, the main contribution of this paper is the discussion of the main challenges to design dedicated hardware for video encoding and decoding considering modern standards. For that, the main tools and techniques used in these encoders are briefly discussed and some published solutions with dedicated hardware systems are presented. Considering the space limitation, only a few of the most representative works are presented and discussed.

This work focuses on three modern video coding standards: High Efficiency Video Coding (HEVC), AOMedia Video 1 (AV1) and Versatile Video Coding (VVC). HEVC is an ISO and ITU-T standard launched in 2013, which duplicates the compression rates for the same video quality when compared to its predecessors. AV1 and VVC are the current state-of-the-art standards, launched in 2018 and 2020, respectively. VVC is the successor of HEVC, and it was developed through a cooperation between ITU-T and ISO ex-

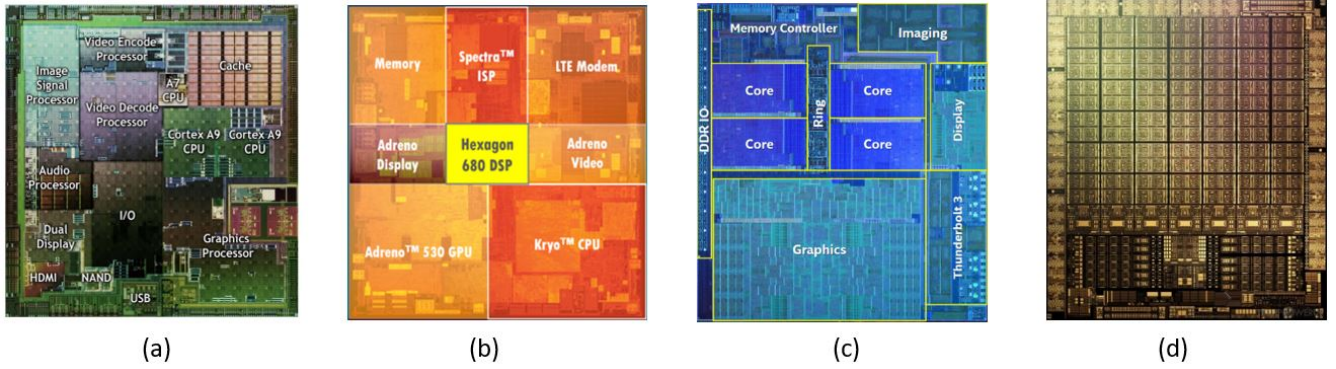


Fig. 1: Examples of chips with dedicated codecs: (a) SoC NVIDIA Tegra 2 [4], (b) SoC Qualcomm Snapdragon 820 [5], (c) GPP Intel i7-1065G7 [7] and (d) GPU GeForce RTX 30 Series [9].

perts. AV1 is a format based on Google's VP9/VP10 [17] and aggregates technologies of other codecs, like Thor from Cisco [18] and Daala from Mozilla [19]. AV1 is a video format developed by the Alliance for Open Media (AOM), a consortium which includes some of the biggest technology companies with the aim of developing a royalty-free and efficient video encoder [20].

The design of high-throughput and low-energy solutions for the highly complex video coding algorithms is, itself, the main challenge in this scenario. When considering battery-powered devices, an additional and particularly important challenge is related to energy consumption, since the high throughput required to process high-resolution videos in real time must be reached at an energy consumption that is as small as possible. These challenges will be better discussed in the next sections, but it is clear that smart dedicated solutions are needed to overcome them in an efficient way.

The rest of this paper is organized as follows. Section II briefly describes the video coding and decoding process. Sections III to VII present the challenges related to the inter-frame prediction, intra-frame prediction, transforms and quantization, entropy encoder and in-loop filters, respectively. Section VIII presents the challenges related to memory communication. Finally, Section IX presents the conclusions of this article.

II. VIDEO CODING AND DECODING PRINCIPLES

A digital video is a sequence of static images, called frames. The frames must be captured and displayed in at least 24 frames per second, which is called the frame rate. The frame size is called spatial resolution and it is defined by the number of pixels in the horizontal and vertical dimensions. Each pixel has three color information, and each of them is called as sample. Thus, each frame is a combination of three matrices of samples, one for each color element. Current video coding standards use the YCbCr color space to represent the pixels. The first frame matrix Y contains the luminance information (brightness), whereas the Cb and Cr matrices contain the blue chrominance and the red chrominance information. Current video encoders process luminance and chrominance samples using specialized tools [21].

To be encoded, the frames of a video are divided into blocks for processing, which are the basic partitioning units [22]. Each block can be further divided to improve the co-

ding efficiency according to specific characteristics of each frame region. The most recent video coding standards support large block sizes, which are more adequate for higher resolutions. Besides, they support a large set of block subdivision formats, providing a flexible partitioning structure that can adapt itself for different types of content. All these features allowed an important increase in coding efficiency, but resulted in a significant computational cost increase, especially when considering high resolutions and frame rates.

The video coding process of modern standards follows the block-based hybrid scheme shown in Fig. 2. Even though there are differences between the operations defined in each standard, the basic encoding steps are the same.

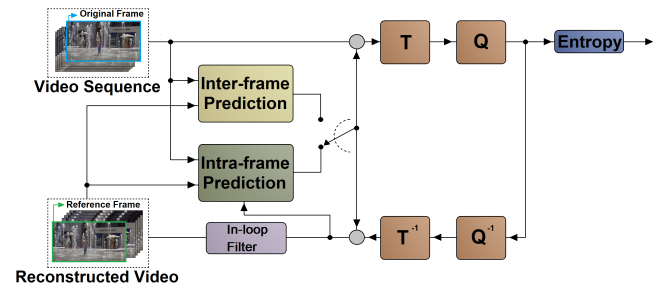


Fig. 2: Main steps of current video encoders.

The inter-frame prediction is responsible to identify and reduce the temporal redundancy between neighboring frames of a given scene. The motion estimation (ME) tool compares each block of the current frame with the previously encoded frames to find the most similar block in the references (the best match). This is done using a search algorithm and the ME results in a motion vector (MV), which indicates the position of the best match in the reference frame. Then, the motion compensation (MC) uses the MV to assemble the predicted frame through the copy of the reference frame blocks to the encoded frame buffer. The inter-frame prediction of current encoders supports other tools, like fractional motion estimation (FME) (which uses interpolation to generate fractional motion vectors) [23], affine motion compensation (which predicts other movements than translational, including scaling, rotation, shape changes and shearing, using more MVs) [24], skip (with reuse of the prediction information from neighboring blocks) [25], and others.

The intra-frame prediction is responsible for reducing the spatial redundancy in a video, using only the information

from the current frame. There is a wide variety of intra prediction tools in current video coders, including an expressive number of directional modes (which interpolates the neighbor samples of a block in a variety of directions), DC mode (which use only the average value of the references to generate the prediction), planar mode (the prediction is generated with a horizontal and vertical smooth gradient) [26], sub-partitions (which divides the block to get closest references) [27], and others. The prediction from the inter or intra-frame steps is subtracted from the original block, generating the residues that are processed by the next encoding steps.

The transform tool (T module in Fig. 2) is applied over the residues to transform them from the spatial domain to the frequency domain. This process is done to better explore the behavior of the human visual system (HVS). Current encoders support a variety of transforms, like the discrete cosine transform (DCT), discrete sine transform (DST), asymmetrical discrete sine transform (ADST), and others [28] and allow the use of different transforms in the horizontal and vertical directions [29].

The quantization step (Q module in Fig. 2) is used to remove or attenuate transform coefficients that are less relevant to the HVS [21]. This is a lossy operation, and the strength of the quantization defines the reached compression rate and the quality losses. Thus, although simple, it is a especially important operation. The encoders use a parameter called quantization parameter (QP) in HEVC and VVC and control quality (CQ) in AV1 to define the quantization strength. Typically, the quantization outputs are sparse matrices.

The entropy encoding is the final step of the encoder process and it applies lossless variable length coding algorithms intending to efficiently compress the sparse matrices generated by the quantization step. The main idea is to represent as much data as possible in as lower number of bits as possible. Currently, the encoders use variations of context adaptive arithmetic coding [30] at this step.

Since quantization is a lossy process and as both the encoder and decoder must use exactly the same references to avoid quality degradation, the inverse quantization (Q^{-1} module in Fig. 2) and the inverse transform (T^{-1} module in Fig. 2) are also present in the encoder.

The last encoder step is the in-loop filtering. These filters aim at improving the subjective video quality by reducing or eliminating artifacts generated during the encoding process, such as blocking, ringing and blurring effects [25]. Besides the increasing in the subjective video quality, the current in-loop filters also increase the encoding efficiency [31]. Current encoders use more than one filter, like the deblocking filter (DBF) [14], the switchable loop restoration filter (SLRF) [32], the sample adaptive offset (SAO) [33], and others.

The video decoding process is presented in Fig. 3, with the inverse process of the encoder. Notice that the decoder is much simpler than the encoder, since all encoding decisions are signaled at the bitstream and the decoder only must follow the already decided processes and coding modes.

An interesting observation at this point is that video coding standards define the bitstream syntax and the decoder tools. This means that the decoder must support all combination of tools defined by the video encoding format, but the

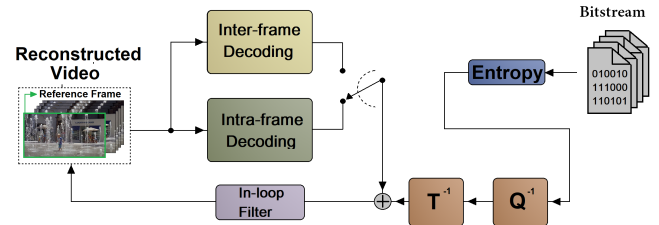


Fig. 3: Main steps of a current video decoder.

encoder can be simplified (removing some encoding tools, for example) as long as the bitstream syntax is respected.

III. INTER-FRAME PREDICTION

The inter-frame prediction provides high compression gains in the encoding process, but it is the coding step that also requires the highest computational effort and the highest memory bandwidth in current encoders. The main features that cause the high computational cost are: the large number of supported block sizes, the number of reference frames, the number of required interpolation filters in FME and MC, the new tools for non-translational motion, among others. The high memory bandwidth required is function of the very large number of reference blocks that must be compared with each current block to define the best match. Since many frames can be used as references and as video resolutions have increased significantly in the last years, the volume of data required by the ME of current encoders is extremely high.

The HEVC codec allows the inter prediction to use five reference frames to search for the best block, three previous frames and two future frames. The number of block sizes supported are 25, from 64x64 to 4x4 pixels. The HEVC FME has a precision of 1/4 of pixel for luminance and 1/8 of pixel for chrominance, using a set of three FIR interpolation filters, with eight or seven taps. HEVC defines two MV modes: advanced motion vector prediction (AMVP) and merge mode. AMVP can combine the motion information from the current block and adjacent blocks. Merge mode allows a block to reuse the MVs from neighboring blocks. There are several hardware implementations targeting the HEVC ME in the literature, like [34], [35], [36], [37], [38] and [11].

The AV1 inter prediction uses up to seven reference frames, four previous frames and three future frames. The number of supported blocks are 22, varying from 128x128 to 4x4 pixels. The AV1 FME precision is of 1/8 pixel for luminance and 1/16 pixel for chrominance. Ninety FIR filters are used in the interpolation process, which vary from eight to two taps. AV1 also defines the novel warped motion compensation, which explore affine transforms to map other movements besides translation, including scaling, rotation, shape changes and shearing. Other innovation is the advanced compound prediction (ACP), which allows four modes: compound wedge prediction, difference-modulated masked prediction, frame distance-based compound prediction and compound inter-intra prediction. These four modes allow the combination of two different predictions to generate a more efficient prediction. In this case, even a combination between intra and inter prediction is allowed. AV1 also defines the overlapped block motion compensation (OBMC), devised to minimize prediction errors near the block borders. A few

works in the literature present hardware designs for AV1 inter prediction. In [39], the authors present a hardware for the MC interpolation filters, whereas in [13] and [40] the authors focus on the AV1 FME interpolation filters.

The VVC inter prediction can use up to 16 reference frames: eight past frames and eight future frames. VVC supports 28 different block sizes in inter prediction, from 128x128 to 4x4 pixels. The FME precision for luminance and chrominance is of 1/16 and 1/32 pixel, and 15 FIR filter are used, varying from 4 to 8 taps. As in AV1, VVC also supports the affine transformations through the affine motion compensation (AMC) tool. VVC defines the combined inter and intra prediction (CIIP), used to improve the prediction quality, exploring the same idea of the ACP compound inter-intra prediction of AV1. An important novelty introduced in VVC is the geometric partitioning mode (GPM), which allows an additional block division in a variety of diagonal and asymmetrical options. Some hardware solutions have been published in the literature for the VVC inter prediction, such as [12], [41] and [42].

Some general characteristics, such as the increased number of block sizes and supported reference frames, together with the higher spatial resolutions and frame rates, contribute to the complexity increase of inter-frame prediction in the most recent video coding standards. Table I summarizes the inter prediction characteristics present in each standard. From HEVC to VVC, it is possible to perceive a significant increase in the number of tools and features in inter prediction, mainly related to the the number of block sizes and reference frames. In addition, novel and complex modes are tested, such as the Affine/Warped mode in VVC and AV1.

Table I.: Inter prediction in current video codecs

Tool/Feature	HEVC	AV1	VVC
Block Sizes	25	22	28
Reference Frames	5	7	16
FME Precision	1/4	1/8	1/16
Filters in FME	3	90	15
Affine/Warped	No	Yes	Yes
Inter-Intra Prediction	No	Yes	Yes
Other novel tools	-	ACP, OBMC	GPM

Works targeting dedicated hardware designs for the inter prediction explore several techniques to obtain gains in coding efficiency, energy consumption, required area, high throughput and memory bandwidth, such as: multiplierless implementations [11], [12], [13], [37], [39], [40], [38]; approximate computing solutions [37], [13], [42]; reuse of common subexpression [11], [39], [12]; parallelism exploration [11], [37], [39], [40]; and hardware reconfigurability [41].

There are many works in the literature targeting hardware designs for the HEVC inter prediction. For example, the work [34] presents a highly parallel motion estimation architecture for the encoder. The proposed architecture has 16 processing units operating in parallel to calculate the sum of

absolute difference values of all possible prediction block sizes. At 720 MHz clock frequency, the proposed architecture processes 2K (1920x1080) resolution at 30 fps with a search window of 55x55 pixels using the full search algorithm. Moreover, resolutions such as 4K (3840x2160) at 30 fps can be achieved using different algorithms [34].

Other example is the work [37], which presents a low-power and memory-aware hardware for the HEVC FME interpolator, with two novel hardware designs for the interpolation filters. These solutions exploit the usage of approximate computing at both algorithmic and data levels, leading to a reduction in dissipated power and memory bandwidth. The proposed design is capable of real-time interpolation of Ultra High Definition (UHD) 4K and 8K videos when synthesized using a 40 nm standard-cell library, with a power dissipation ranging from 22.04 to 62.06 mW [37].

As AV1 is a more recent encoder than HEVC, only a few works targeting hardware designs for the inter prediction are available in the literature. The work [13] presents an approximate solution for the AV1 FME interpolation filters based on the approximation of the original filter coefficients intending to generate more hardware-friendly coefficients. The approximated version was designed in hardware and can achieve real-time interpolation for UHD 8K videos at 30 frames per second when synthesized using 40nm TSMC standard-cells technology. The designed architecture dissipates 26.79 mW, which represents more than 80% power reduction when compared to the original precise solution. The approximation implies in a small average coding efficiency degradation of 0.54% in BDBR [13].

VVC, as the newest coding standard, has only a few works with dedicated hardware solutions published in the literature. Considering the inter prediction, [42] proposes the use of approximated filters for the VVC fractional interpolation. When compared with the original solution, this work implements 14 3-tap filters and one 4-tap filter, instead of the 15 8-tap filters defined in VVC. The architecture was synthesized for the Xilinx Virtex-7 FPGA and shows a power dissipation 40% lower than the original version, being able to process 47 fps of Full HD videos.

Based on these works for the inter-frame prediction of HEVC, AV1 and VVC, one can conclude that this stage demands a continuous research effort, since its high complexity and its high memory bandwidth are real bottlenecks in video coding systems. These challenges are especially critical for AV1 and VVC codecs, since they were recently introduced and bring novel and complex inter prediction tools, besides the support for more block sizes and more reference frames in a scenario with increasing resolutions and frame rates.

IV. INTRA-FRAME PREDICTION

The intra-frame prediction is another important tool of current video codecs. The addition of novel block sizes and new intra modes and tools are responsible for most of the increased complexity for this module. Table II summarizes the main characteristics of the intra-frame prediction for HEVC, AV1 and VVC, which are discussed as follows.

The HEVC implementation of the intra prediction employs directional and non-directional predictions. On HEVC, there are 35 intra modes that can be applied to the

Table II.: Intra prediction in current video codecs

Feature/Tool	HEVC	AV1	VVC
Block sizes	5	19	25
Intra modes	35	62	67
Sub-partition	No	Yes	Yes
Other novel tools	-	Paeth, Cfl Smooth	WAIP, SCC MRL, MIP

five squared block sizes, from 64x64 to 4x4. Besides the non-directional DC and planar modes, HEVC presents 33 possible directional modes for this module. There are plenty of works proposing dedicated hardware architectures targeting the HEVC intra prediction, such as [43], [44] and [45].

The AV1 intra prediction comprises 62 modes. From these modes, six of them are non-directional: DC, Paeth, Smooth Vertical, Smooth Horizontal, Smooth and Recursive-based-filtering. The Paeth prediction is a novel tool introduced in the AV1 format, which employs the gradient function and considers three samples as reference to make the prediction. The three Smooth modes are similar to the HEVC planar mode and apply a linear interpolation to predict smooth surfaces, using filtered samples from reference arrays. Recursive-based-filtering (RBF) is another AV1 novelty, which aims at alleviating decaying spatial correlations as the predicted block positions get further away from the reference samples, using sub-partitions within the block. The other 56 modes are directional, composed by eight nominal angles that can be slanted through seven variations. AV1 also introduced a chrominance-from-luminance (CfL) mode, which can predict chrominance samples from reconstructed luminance samples. Currently, a limited number of works in the literature present hardware architectures targeting this module for AV1 [46], [47] and [48].

Regarding the state-of-the-art VVC intra-frame prediction, 67 prediction modes are available, reducing the prediction error but requiring a much higher computational effort. VVC included the planar and DC modes already available in HEVC as the non-directional prediction options. The other 65 modes are directional, using different angles to predict the current block [49]. Additionally, there are other innovations, such as: wide-angle intra prediction (WAIP), used to apply directional intra modes to non-square blocks; multiple reference line prediction (MRL), allowing the use of more reference lines; intra sub-partitions (ISP), applied to explore correlations among intra-block samples; and matrix-weighted intra prediction (MIP), which performs the prediction through matrix multiplications and sample interpolation. VVC intra prediction is applied to blocks of 64x64 pixels or smaller. It can also be applied to rectangular blocks, which was not supported by HEVC [50]. Thus, a total of 17 different block sizes are supported by the VVC intra prediction. Although this standard presents an increased complexity for intra operations, there is only one work that proposes specific hardware architectures for this module [51].

The intra-frame prediction module presents many challenges in novel video coders, especially related to the great

amount of possible block sizes and intra modes available. Although there are studies proposing new hardware architectures for this module, there are still many research opportunities for works aiming to reduce the intra prediction complexity. In this scenario, hard challenges with respect to architectural design to support all possible intra-frame prediction features are imposed, such as providing high throughput while respecting power and memory bandwidth budgets.

For HEVC, novel tools and architectures have been widely proposed, incorporating a variety of techniques and heuristics. For example, in [44], a hardware-friendly internal mode decision algorithm was proposed, significantly reducing the number of necessary arithmetic operations. This strategy allows an architectural design that reaches real-time video encoding for UHD 8K resolutions at 120fps, with a power consumption of 363mW. In [45], the authors present a new algorithm, as well as a hardware architecture capable of supporting it. This strategy takes advantage of five different intra techniques and this solution can process 2160p resolution at 30fps, with a power consumption of 273mW.

For AV1, a few works are available in the literature focusing in hardware implementations. In [46], a parallelized architecture is proposed for AV1 intra prediction, applying edge filtering and upsampling to blocks, while supporting all directional modes available. The proposed architecture can process 1080p resolution at 60fps with a power consumption of 382.08mW. On the other hand, [47] proposes a hardware architecture focusing on the non-directional modes, achieving a high throughput for UHD 4K video sequences, and supporting all 19 available block sizes for this module. This solution processes up to 30 fps at 4K resolution, with a power consumption of 65.5mW.

In VVC, intra prediction is the most compute-intensive step, according to [50] and [52]. Since this standard is still very recent, a few works focus on hardware solutions for the VVC intra prediction step. In [51], the authors propose a FPGA architecture for this module, supporting square blocks from size 4x4 to 32x32, using only two DSP blocks and two adders to implement the operations of intra modes. The architecture processes 34fps at HD 1080p resolution.

Considering all the presented discussion regarding the intra-frame module, plenty of challenges regarding the novel coding tools introduced by the most recent video coding standards emerged, requiring the development of even more efficient algorithmic optimization strategies. In this sense, machine learning-based strategies have attracted the focus of researchers, being a promising solution to achieve better complexity reduction levels than heuristics-based algorithms. Furthermore, dedicated accelerators to meet real-time processing focusing on ultra high resolutions are still required and must explore the parallelism intrinsic to the algorithms and minimize memory communication issues.

V. TRANSFORMS AND QUANTIZATION

Both the transform and quantization modules are part of the encoding loop and prepare residual data for the entropy coding process. In recent video coding standards, which allow the use of several block sizes and formats, as well as many different prediction modes, the number of residual blocks to be processed by the transform and quantization

modules is huge. If optimal or near-optimal mode decision is a goal, the throughput of these modules become a significant challenge, since they need to provide all the quantized coefficient blocks for the entropy coding module based on a large amount of block sizes/formats and prediction mode combinations [22]. The main differences of these modules for HEVC, AV1 and VVC encoders are presented in Table III and are discussed in the next paragraphs.

Table III.: Transforms and quantization in current video codecs

Feature/Tool	HEVC	AV1	VVC
Block sizes	4	17	19
Transform Types	3	16	8
Different H/V	No	Yes	Yes
Quantization Parameters	52	256	63

Besides that, recent video coding standards provide the flexibility of using different 2D transforms and even combinations of two or more 1D transforms for the same residual block. Although the Discrete Cosine Transform type II (DCT-II) is the most popular solution due to its superior capability of balancing coding efficiency and computational cost [53], several other transforms, such as the discrete sine transform (DST) and the asymmetric discrete sine transform (ADST) have been employed in recent standards. The possibility of evaluating different transform types and combinations for all candidate blocks increases even more the required throughput of the transform and quantization steps, especially at the encoder side. Thus, efficient hardware optimization techniques for the transform and quantization modules are essential in current video coders.

In HEVC, the DCT II transform processes blocks between 4x4 and 32x32 samples. Specifically for 4x4 blocks, an alternative integer transform derived from the DST is applied to the luminance residual blocks generated from intra prediction modes [29]. Also, a transform skip can be applied, which avoids the transform process. Regarding the quantization module, HEVC employs a scalar quantization with a QP that ranges from 0 to 51. Some works proposing hardware solutions for the HEVC transforms can be found in [54], [55] and [56]. Some works have also been published focusing on hardware for the quantization module, such as [57] and [58].

The AV1 transform module supports block sizes from 4x4 up to 64x64, including rectangular blocks. Depending on the prediction mode, a combination of two 1D transforms can be applied to the residual block, one horizontally and the other vertically. To improve coding efficiency, a recursive transform block partition was introduced in AV1, which aims to search for stationary regions located only in inter-predicted blocks. AV1 introduces sixteen 2D combinations of four 1D transforms: the DCT II, the ADST, the flipped ADST (flipADST), and the identity transform (IDTX). In addition to the main transform, AV1 also introduces a non-separable secondary transform, which leads to better compression efficiency for directional texture patterns [59]. In this format, a scalar quantization is also employed and the quantization

parameter value ranges between 0 and 255. AV1 also includes 15 sets of predefined quantization weighting matrices, where the quantization step size for each frequency component is scaled in a different way. To the best of the authors' knowledge, there are no works published in the literature focusing on hardware architectures for the transforms and quantization modules of AV1.

One of the main innovations of VVC is the introduction of the multiple transform selection (MTS) tool for both intra-predicted and inter-predicted blocks. The transforms included in MTS are the DCT II, the DCT VIII and the DST-VII. The use of DCT and DST allows a separable transformation, which means that the block transformation can be applied in the vertical and horizontal directions separately, similarly to what happens in AV1. However, when the DCT-II is selected by the encoder, it is always applied in both directions. As in HEVC, VVC also includes the possibility of skipping the transform process. As in AV1, VVC presents a secondary transform called low frequency non-separable transform (LFNST), which is only applied over 4x4 or 8x8 blocks. At the moment, there are a few works found in the literature with hardware proposals for the transform and quantization modules of this standard, such as [60], [61], and [62].

The introduction of new transform types, as well as the increase of block size possibilities and prediction modes allowed in HEVC, AV1 and VVC incur in the rise of important challenges related to the transform and quantization steps, especially in terms of throughput. Notice that hardware solutions for these modules in the encoder need to process not only the residual block resulting from the chosen block size/format and the chosen prediction mode. Instead, for optimal or near-optimal coding efficiency, they are required to process several combinations between block sizes/formats and prediction modes. Supporting many types of transforms also has several consequences related to memory allocation, since temporary results from different residual block candidates need to be stored, processed with different transform types and quantized. Therefore, providing high-performance design under the hardware constraints of the target device is a crucial issue [63].

Most works that propose hardware solutions for the transform and quantization modules focus on optimizations using techniques to reduce the number of calculations and hardware reuse. Some of these works also employ hardware approximation approaches, such as [63]. The work in [64] employs resource reutilization techniques, because the transforms present very similar operations for different block sizes, so it is possible to share calculations between them.

A recently published work featured in [54] presents a reconfigurable architecture for the DCT that allows the reutilization of different transform sizes for the HEVC encoder. The architecture was synthesized for the Stratix II FPGA family and achieved a processing rate of 558.6 to 4468.8 Mpi-pixels/sec for ultra-high-definition videos (3840x2160) in real time, with a frame rate of 45 fps up to 359 fps.

In [63], the authors discuss the hardware implementation of an approximated MTS module for the VVC standard. Two solutions are proposed: the first one proposes an unified and pipeline-efficient architecture for direct and inverse DCT-II

for block sizes of 4x4, 8x8, 16x16 and 32x32 samples with low computational complexity and logical resource allocation, whereas the second solution presents a 2D implementation of the forward and inverse DST-VII and DCT-VIII transforms with an approximate design through adjustments stages. Synthesis results show that the design supports 2K and 4K videos at 377 and 94 frames per second, respectively, while using only 18% of the Adaptive Logic Modules (ALM), 40% of registers and 34% of DSP blocks.

Since nowadays most video coding standards usually employ a coding scheme that uses the DCT family, it is very common to find works in the literature with hardware solutions for this type of transform only. As other transform types have only been introduced in VVC and AV1 more recently, works that focus on different transform cores are still rare.

VI. ENTROPY ENCODER

The increased complexity of the entropy encoder is related to the types of compression algorithms supported and to the increase in resolution and frame rate of current videos. The coding efficiency advantages presented by these types of algorithms provide a reduced bitstream generation, but require an increased computational effort.

The HEVC entropy coder uses the context-adaptive binary arithmetic coding (CABAC). The CABAC encoding process is composed by three basic stages: binarization (BI), context modeling (CM), and binary arithmetic encoding (BAE). The input data comes from all previous video encoding steps, named as syntax elements (SE). The BI consists of a mapping of integer values into a sequence of bits that represent the original value, resulting in binary symbols, called bins. There are two types of bins: regular and bypass. Bypass bins have fewer data dependencies among them and go straight to the last stage. The regular bins pass through the second stage. This mapping reduces the size of the alphabet of symbols simplifying the costs to the second stage and facilitating the BAE, considered the most critical stage from CABAC. The CM stage calculates the probability estimation of regular bins based on some specific context. BAE compresses the bins into a bitstream based on the probabilities selected in the previous stage and then the data finally is ready to be transmitted or stored [30]. There are several works in the literature implementing dedicated hardware for the HEVC entropy coding, especially for the BAE stage: [30], [65], [66], [67], and [68].

The AV1 entropy coder uses an adaptive multi-symbol arithmetic encoder applied symbol by symbol to compress SEs. Each SE in AV1 is a member of a specific alphabet of N elements and the entropy context consists of a set of N probabilities. A cumulative distribution function (CDF) is responsible to store that probabilities of 15-bit precision length. The arithmetic coding uses the CDF to compress symbols and provides better results than the BAE from HEVC, which uses probabilities from the CM [25]. There are no works in the literature presenting dedicated hardware solutions for the AV1 entropy coder until the moment this paper was written.

Similarly to HEVC, the VVC entropy coder uses CABAC for all low-level SE. The non-binary SE are mapped to binary codewords in the BI stage. The bins of both binary SE and codewords for non-binary data are coded using the BAE.

The BI stage and context modeling are still used and have a significant impact on coding efficiency. A regular mode is used where the bin strings are coded by adaptive probability models, and a less complex bypass mode is used with fixed probabilities of 1/2. These modes are also called contexts and the assignment of probability models to individual bins is referred as a CM. The process is finalized with the BAE [69], which generates the bitstream. The current literature does not present any papers focusing on dedicated hardware for the VVC entropy coding.

The main challenge of entropy coding in HEVC, AV1 or VVC is the fact that the arithmetic coding is a sequential algorithm that leads to a strong data dependency. This makes the parallelism exploration a hard work, especially for the BAE stage. The difficulty of a hardware design approach is associated to these intrinsic dependencies between bins during the processing. Nevertheless, there are successful research works pointing out innovations and increasing throughput for the CABAC module, such as [65].

Even though there are several works that propose hardware architectures for the HEVC entropy coding, some of which have been already discussed, there are still no works for this module of VVC and AV1 coders. At this point, it is possible to highlight that most entropy coding hardware implementations found in the literature have the goal of overcoming data dependencies and exploring parallelism to provide high throughput, thus allowing the processing of high-resolution videos in real time.

In general, the works focusing on HEVC present hardware solutions for the BAE stage. The paper in [30] presented a novel scheme to process multiple bypass bins, which is able to process 8K UHD videos in real time with a low-power approach, which groups bins of certain types and turns off the parts of the architecture that are not required to process these bins. The clock gating and operand isolation techniques were also used and provided power savings [30].

A multicore-based solution that targets high throughput processing is presented in [65], where a novel scheme is proposed to support entropy coding with a continuous processing rate. This hardware architecture deals with the intrinsic dependencies of the SEs in a way to allow the delivery of multiples SEs per clock cycle for any given CABAC block. The solution [65] presents the best tradeoff between bins per cycle and maximum frequency in comparison to others CABAC designs found in the literature. The work [66] presents a pipelined CABAC architecture to increase the reached operating frequency. Finally, the work [67] presents a CABAC encoder architecture with a four-stage pipeline for the BAE core using the bypass bin splitting technique, thus achieving a high throughput.

All these solutions are aligned with the current demand to process high-resolution video in real time. In general, the proposed designs focus on improving the throughput to process multi-bins and to increase the number of bins processed per cycle. There are still many challenges to be overcome in the entropy coding step, mainly for the AV1 and VVC standards, since the current literature does not present any hardware designs targeting them.

VII. IN-LOOP FILTERS

The in-loop filters complexity has also increased in the most recent codecs. The in-loop filters improve image quality by reducing or eliminating artifacts that cause image degradation and they also increase the coding efficiency by improving reference images used in the prediction steps. Each video coding standard or format specifies its own in-loop filters.

HEVC specifies two in-loop filters: the deblocking filter (DBF) and the sample adaptive offset (SAO), which is a deringing filter. The HEVC DBF is applied over 8x8 blocks and the filter is composed of three FIR filters: a normal filter (4 taps) and a strong filter (5 taps) for luminance and a Chrominance filter (3 taps) for chrominance [14]. The SAO filter comprises an offset derivation stage and a filtering stage. The first stage collects required statistics information from original and reconstructed blocks and the filtering stage performs offset filtering according with the offsets and chosen filter types [33]. There are several works in the literature implementing dedicated hardware for the HEVC DBF, like [70], [31], [14], and for the HEVC SAO, like [71] and [33].

AV1 presents three in-loop filters: the DBF, the constrained directional enhancement filter (CDEF) and the switchable loop restoration filter (SLRF). DBF is the first filter applied and defines four FIR filters with 13, 7, 5, and 4 taps, working separately for luminance and chrominance samples [72]. The CDEF is applied after the DBF and it is a deringing filter. The CDEF is a non-linear directional low pass filter [73] composed of two main tools: the direction search (DS) and the non-linear low-pass filter (NLLPF). The DS identifies each block direction and the NLLPF is a 12-tap filter where the taps are defined according to the DS direction [74]. The SLRF is applied after the CDEF and it is a deblurring filter that works over 64x64, 128x128 or 256x256 blocks [25]. The SLRF is composed of two filters: the dual self-guided filter (DSGF) and the symmetric normalized Wiener filter (SNWF). The DSGF operates based on two self-guided images [32] and the SNWF consists of a 7-tap linear filter using different coefficients based on the sample characteristics. There is only one work published in the literature that presents a hardware architecture for the AV1 DBF [72]. There are also two works implementing the CDEF hardware [75], [74]. Finally, the literature does not present works with hardware designs for the SLRF.

VVC presents four different in-loop filters: DBF, SAO, adaptive loop filter (ALF) and cross-component adaptive loop filtering (CC-ALF), which are applied in this order. The DBF is conceptually similar to the HEVC DBF but presents several enhancements, and SAO is identical to the one used in HEVC. ALF and CC-ALF operate in parallel. Both are adaptive filters and are designed to enhance the reconstructed signal, based on Wiener-filter encoding approaches [76]. The ALF operations are composed by a classification process and a filtering process used as an optimal linear filter, where each 4x4 block is analyzed and classified into 25 categories according to directions and activity of local gradients. The CC-ALF uses the correlation between luminance and chrominance samples and is applied only to chrominance samples, where a Wiener filter is used to reduce the mean square

error (MSE) between the original and the reconstructed samples [76]. There are no works in literature that present dedicated hardware solutions for the VVC in-loop filters.

In general, the in-loop filters are designed exploring: (i) the use of parallelism, to provide the required high throughput; (ii) low-power techniques, to support battery-powered devices; (iii) the use of common sub-expression sharing, as a strategy to reduce the number of operations and area consumption; (iv) multiplierless solutions, to decrease the amount of required computational resources; and (v) dedicated memory implementations, to reduce the number of data accesses in the main memory, thus enhancing timing efficiency. Examples of these solutions are available, respectively, in [74], [75], [14], [70], and [31].

For HEVC, an example of a highly efficient DBF architecture is presented in [31]. This design presents a solution to reduce the number of data accesses to the external memory implementing its own data structure, enabling high processing throughput and low complexity. It was implemented using the TSMC 90nm standard cell library and achieved a throughput of 60 fps for a resolution of 4096x2048 pixels, under a frequency of 100MHz, occupying 466.5 Kgates. For the HEVC SAO, a hardware design targeting the encoder side is presented in [71], which comprises the main SAO stages for the classification methods, statistical processing to generate the offsets and filtering operation. The proposed design targets the processing of high resolution video (1920x1080 and 3840x2160) in real time, when synthesized for an Altera Stratix V FPGA. The architecture uses 8,040 ALUTs of the target device and is capable of processing 44 QFHD fps when running at 364MHz.

The AV1 CDEF architecture proposed in [74] targets the decoder side and the proposed solution reaches real-time processing of 4K UHD videos at 60fps under a frequency of 93 MHz. It was implemented using the 40nm TSMC library, consuming an area of 185.36 Kgates with a power dissipation of 43.29 mW. The AV1 DBF hardware architecture targeting the decoder presented in [72] reaches real-time processing of 4K UHD videos at 60fps under a frequency of 16.2 MHz. It was implemented using the 40nm TSMC library consuming an area of 39.35 Kgates with a power dissipation of 3.96 mW.

As cited before, there are no works in the literature reporting hardware designs for the VVC in-loop filters. A clear challenge for hardware implementation of the in-loop filters is how to connect them all, since the filters work in sequence with different goals, operation modes and requirements. For example, the number of block sizes and formats processed by the filters have also increased in recent standards and they differ from one filter to another. Another challenge is the large number of samples to be filtered, requiring parallelism exploration to reach the required throughput. The filters also demand data accesses to/from memory and need to store samples and intermediate results, which is an aspect that must be considered when designing this module.

VIII. MEMORY ISSUES

The memory usage of a computing system is responsible for a large portion of its energy consumption. Therefore, it is essential that hardware solutions for implementing video codecs consider the memory infrastructure of the architecture.

In this section, an evaluation of the memory requirements for each encoder module will be presented. This analysis emphasizes the state-of-art VVC and its predecessor, HEVC, although most results can be generalized to video coders based on the hybrid video coding model, including AV1.

The inter-frame prediction is responsible for most of the memory accesses performed during the encoding process. This overhead in memory requirements can be explained by the exploitation of reference frames during this encoding step, which are stored in memory buffers. Although the memory usage of the inter prediction module increased in modern video coders, it seems to be proportionally less representative than it was on previous standards, which can be explained by the overhead in memory accesses led by other novel tools included in other parts of the encoder.

For a comprehensive quantitative evaluation of memory requirements of each analyzed video coding module, we selected works [77] and [78], which evaluated the memory requirements of the inter prediction module on two modern video encoders: HEVC and VVC. Considering the state-of-the-art VVC, the inter-frame prediction module is responsible for up to 85% of the encoder memory usage, achieving an absolute average increase of 3.5x when compared to its predecessor, the HEVC. The adoption of larger block sizes on VVC (bigger than 64x64, which were not available on HEVC) seems to be one of the main factors responsible for this increase, reaching up to 23.3% of the inter prediction memory requirements. The same trend can be derived for AV1 codecs, since larger block sizes and more flexible frame partitioning structures are also defined by this standard. Novel inter prediction tools of VVC, such as Affine and GMP, were also evaluated regarding their memory accesses. However, they do not appear to represent a major overhead in the memory requirements of this module.

As the most memory-intensive task present in a video encoder, the inter-frame prediction is the main target of memory architecture optimizations in terms of its on- and off-chip data storage. Most of such research efforts are directly related to the Decoded Picture Buffer storage, which stores the past reconstructed frames (called reference frames). When HEVC took place as the state-of-the-art technology, it aggravated the memory issues from the previous standards, posing new challenges for the memory infrastructure. Besides new approaches based on the previous encoders, HEVC brought light-weight parallelism support to accelerate the encoding process, raising new requirements to the memory infrastructure, which must have to support larger memory bandwidth due to the simultaneous accesses of multiple processing units. In this context, several works exploit novel memory approaches to handle these challenges, adopting multi-level dedicated memory hierarchy [79], [80], emerging memory technologies with low-power features [81] and approximate storage [82].

Memory design challenges for AV1 and VVC inter prediction continue to be aggravated due to the more complex coding tools introduced by these standards, but also due the increase in video resolution and frame rate. Thus, novel solutions are mandatory to meet performance and energy requirements imposed by current multimedia applications.

The intra prediction module does not present a major memory requirement overhead when compared to the inter-frame prediction. This behavior is expected since the intra prediction uses only information from the current frame to perform its operations, without the necessity to load reference frames from memory buffers. This module is also affected by variations on the QP, requiring more memory for lower values of this parameter. On VVC, the average representativity of this module ranges from 6% in QP 37 to 13% in QP 22, considering all prediction memory accesses [78]. Intra prediction also presents an increase in memory accesses for modern video coders when compared to previous standards, which can be explained by the addition of more intra modes and more block sizes available in [77].

Concerning the transform and quantization memory accesses, this module presented a significant increase in modern video coders when compared to previous standards. On VVC, the memory usage for these modules ranges from 9% to 17% of all encoder accesses. This module is also strongly impacted by variations on the QP value, reaching 9.9x more accesses on QP 22 than on QP 37 [78]. When compared to its predecessor, VVC also presents a significant memory requirement increase in this module [77]. Since the novel codecs allow the possibility of run-time selection between multiple transforms types and block sizes depending on the residue characteristics, memory-related issues have increased concerns with respect to residue data storage. Although algorithmic optimizations have been proposed to simplify this encoder decision, as in [83], the memory support for the whole residue treatment path has become a major issue.

Regarding memory requirements of the entropy coding, this module does not represent a major overhead for modern video coders. On VVC, this encoding step represents from 2% to 5% of all memory access [78]. This module appears to access more memory for lower QP values, reaching an 8.2x increase from QP 37 to QP 22. The representativity of the entropy memory requirements also seem to remain stable across the different video coders [77].

The deblocking filters also do not appear to play a major role on the encoder memory requirements when compared to other encoding modules. On VVC, this module reaches around 1% to 2% of all accesses [78]. The QP value does not present a critical impact on its memory requirements as well. Previous video coders also seem to achieve similar results [77]. Still, the memory is employed in a wide variety of filter operations and intermediate results, as well as in the great number of input samples required, where there are many opportunities for further optimizations and reduced data accesses in the main memory. In [31], the authors present an efficient VLSI architecture aiming to reduce this module memory accesses, thus improving timing efficiency.

It is possible to observe how the memory usage for inter prediction remains the biggest bottleneck in modern video coders, and there are many research opportunities to optimize its requirements. Although the other modules do not present memory requirements as critical as inter prediction, there is still demand for novel architectures that take memory usage into consideration, enabling a more efficient energy consumption for next-generation video coding standards.

IX. CONCLUSIONS

This work has presented a detailed discussion of the main challenges on designing dedicated hardware for the most modern video encoding standards and formats, presenting also a brief discussion about the encoding tools and some published works focusing on hardware designs for these tools. The video coding formats focused in this work were the ISO and ITU-T HEVC and VVC, and the AOMedia AV1.

We have shown that these challenges are mainly related to the introduction of new coding tools, which significantly increase the computational effort of such video encoders. These new coding tools are necessary to improve coding efficiency (high quality with low bit rate) and enable applications that demand videos with ultra high resolutions, high frame rates and real-time processing. We have discussed dedicated hardware accelerators as the main solution to enable modern video coding systems with high processing rates and low energy consumption.

The main challenges for each video coding step are described in this paper with respect to its methods and previously published works focusing on hardware design. The inter-frame prediction main challenges are related to the high number of block sizes and reference frames supported, the high number of interpolation filters required, the new tools included in the state-of-the-art encoders, and, mainly, the extremely high memory bandwidth required. The intra-frame prediction has also the challenge related with the high number of supported block sizes and the complexity of the novel intra coding modes. The challenges of the transform and quantization steps include the new combinations of different transforms that can be applied, the increased number of supported sizes and the high throughput requirements. The main challenge on the entropy coder is related with the required throughput, since there is an intrinsic difficulty of exploring the parallelism in this sequential tool. Finally, the main challenge of the in-loop filters are related with the increasing number of filters included in the current encoders and their integration. Besides, memory bandwidth is a common challenge for all encoder modules, but mainly for the inter prediction, which is the module that requires the most number of memory accesses in modern encoders. When considering battery-powered devices, another common challenge for all encoder modules is the energy consumption, which must be as lower as possible.

This scenario of multiple challenges are oftentimes contradictory, making the hardware design of dedicated systems an impressive challenging task. For example, the novel encoding tools and partitions intend to increase the coding efficiency, but bring the requirements for a higher computational effort and a higher number of memory accesses. These two requirements avoid the implementation of low-energy solutions. Then, some published solutions sacrifice a part of the coding efficiency to allow for a low-energy implementation. This tradeoff must be carefully evaluated to ensure that the energy gains do not cancel out the coding efficiency gains that the novel encoders provide.

Considering the discussion presented in this paper, one can conclude that there are several research opportunities to be explored in terms of hardware design targeting video co-

ding, mainly for AV1 and VVC. For the VVC encoder there are no published dedicated hardware designs to overcome the challenges on the quantization, entropy coding and in-loop filter steps. For the AV1 encoder there are no published works presenting hardware accelerators for the transform, quantization and entropy coding steps. Even the solutions available for the other steps of these codecs do not solve completely all the challenges pointed out in this paper. Then, there are still many open research opportunities that need to be explored in the near future.

In summary, the new coding tools present in each of the main steps of modern video encoders impose huge challenges for hardware designers, since they will need to come up with solutions for highly complex coding algorithms while still providing high throughput and low energy consumption.

ACKNOWLEDGEMENTS

This investigation was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. It was also financed in part by the Fundação de Amparo à pesquisa do Estado do Rio Grande do Sul – Brasil (FAPERGS), and by the Conselho Nacional de Desenvolvimento Científico e Tecnológico – Brasil (CNPq).

REFERENCES

- [1] “Cisco Annual Internet Report (2018-2023),” <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>, accessed: 2021-30-03.
- [2] “COVID-19 impact: Streaming services to dial down quality as internet speeds fall,” <https://indianexpress.com/article/technology/tech-news-technology/coronavirus-internet-speeds-slow-netflix-hotstar-amazon-prime-youtube-reduce-streaming-quality-6331237/>, accessed: 2020-06-02.
- [3] F. Koushanfar *et al.*, “Processors for mobile applications,” in *Proceedings 2000 International Conference on Computer Design*. IEEE, 2000, pp. 603–608.
- [4] “Snapdragon 865+5G Mobile Platform,” <https://www.qualcomm.com/products/snapdragon-865-plus-5g-mobile-platform>, accessed: 2020-06-04.
- [5] “Qualcomm Details Hexagon 680 DSP in Snapdragon 820: Accelerated Imaging,” <https://www.anandtech.com/show/9552/qualcomm-details-hexagon-680-dsp-in-snapdragon-820-accelerated-imaging>, accessed: 2021-29-06.
- [6] “Snapdragon 820 mobile platform,” <https://www.qualcomm.com/products/snapdragon-820-mobile-platform>, accessed: 2021-29-06.
- [7] “Intel Core i7-1065g7 Benchmarked: Ice lake with Iris Plus Graphic,” <https://www.techspot.com/review/1944-intel-core-i7-1065g7/>, accessed: 2021-29-06.
- [8] “New 10th gen intel® core™ u-series and y-series processors,” <https://intel.com/content/www/us/en/products/docs/processors/core/10th-gen-core-mobile-processors-brief>, accessed: 2021-29-06.
- [9] “GeForce RTX 30 series Graphics Cards: The ultimate play,” <https://www.nvidia.com/en-us/geforce/news/introducing-rtx-30-series-graphics-cards/>, accessed: 2021-29-06.
- [10] “Geforce rtx 30 series gpus: Ushering in a new era of video content with av1 decode,” <https://www.nvidia.com/en-us/geforce/news/rtx-30-series-av1-decoding/>, accessed: 2021-29-06.

- [11] W. Penny *et al.*, "High-throughput and power-efficient hardware design for a multiple video coding standard sample interpolator," *Journal of Real-Time Image Processing*, vol. 16, no. 1, pp. 175–192, 2019.
- [12] A. CanMert, E. Kalali, and I. Hamzaoglu, "A low power versatile video coding (vvc) fractional interpolation hardware," in *2018 Conference on Design and Architectures for Signal and Image Processing (DASIP)*, 2018, pp. 43–47.
- [13] R. Domanski *et al.*, "Low-power and high-throughput approximated architecture for av1 fine interpolation," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021, pp. 1–5.
- [14] R. Palau *et al.*, "Real-time and low-power hevc deblocking filter architecture targeting 8k uhd@ 60fps videos," *Journal of Integrated Circuits and Systems*, vol. 15, no. 1, pp. 1–9, 2020.
- [15] S. Venkataramani *et al.*, "Approximate computing and the quest for computing efficiency," in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2015, pp. 1–6.
- [16] A. Raha, H. Jayakumar, and V. Raghunathan, "A power efficient video encoder using reconfigurable approximate arithmetic units," in *2014 27th International Conf. on VLSI Design and 2014 13th International Conference on Embedded Systems*. IEEE, 2014, pp. 324–329.
- [17] "Chromium open-source browser project, VP9 source code," <http://git.chromium.org/>, accessed: 2019-10-08.
- [18] S. Midtskogen *et al.*, "Integrating Thor Tools into the Emerging AV1 Codec," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sep. 2017, pp. 930–933, DOI: 10.1109/ICIP.2017.8296417.
- [19] J. Valin *et al.*, "Daala: Building a Next-Generation Video Codec from Unconventional Technology," in *2016 IEEE 18th International Workshop on Multimedia Signal Processing (MMSP)*, Sep. 2016, pp. 1–6, DOI: 10.1109/MMSP.2016.7813362.
- [20] "Alliance for Open Media," www.aomedia.org, accessed: 2021-10-02.
- [21] I. E. Richardson, *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons, 2004.
- [22] L. V. Agostini, "Desenvolvimento de arquiteturas de alto desempenho dedicadas à compressão de vídeo segundo o padrão h. 264/avc," 2007.
- [23] "ITU, "Conformance specification for ITU-T H.265 high efficiency video coding Recommendation," <https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=13904&lang=en>, accessed: 2019-10-02.
- [24] P. de Rivaz and J. Haughton, "Av1 bitstream & decoding process specification," *The Alliance for Open Media*, p. 182, 2018.
- [25] J. Chen *et al.*, "Algorithm description for versatile video coding and test model 11 (vtm 11)," *Jt. Video Expert. Team ITU-T SG 16 WP 3 ISO/IEC JTC 1/SC 29/WG 11, 20th Meet. by teleconference*, Oct 2020.
- [26] I. Richardson, "The h.264 advanced video compression standard: Second edition," 04 2010.
- [27] S. De-Luxán-Hernández *et al.*, "An intra subpartition coding mode for vvc," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 1203–1207.
- [28] T. Zhang and S. Mao, "An overview of emerging video coding standards," *GetMobile: Mobile Computing and Communications*, vol. 22, no. 4, pp. 13–20, 2019.
- [29] G. J. Sullivan *et al.*, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.
- [30] F. L. Ramos *et al.*, "Novel multiple bypass bin scheme and low-power approach for hevc cabac binary arithmetic encoder," *Journal of Integrated Circuits and Systems*, vol. 13, no. 3, pp. 1–11, 2018.
- [31] P.-K. Hsu and C.-A. Shen, "The vlsi architecture of a highly efficient deblocking filter for hevc systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 5, pp. 1091–1103, 2016.
- [32] D. Mukherjee *et al.*, "A switchable loop-restoration with side information framework for the emerging AV1 video codec," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sep. 2017, pp. 265–269, DOI: 10.1109/ICIP.2017.8296284.
- [33] M. Mody, N. Nandan, and H. Sanghvi, "Efficient VLSI architecture for SAO decoding in 4K Ultra-HD HEVC video codec," in *2016 29th IEEE International System-on-Chip Conference (SOCC)*, Sep. 2016, pp. 81–84, DOI: 10.1109/SOCC.2016.7905440.
- [34] A. Medhat, A. Shalaby, and M. S. Sayed, "High-throughput hardware implementation for motion estimation in hevc encoder," in *2015 IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2015, pp. 1–4.
- [35] G. Sanchez, M. Porto, and L. Agostini, "A hardware friendly motion estimation algorithm for the emergent hevc standard and its low power hardware design," in *2013 IEEE International Conference on Image Processing*, 2013, pp. 1991–1994.
- [36] W. Penny *et al.*, "Energy-efficiency exploration of memory hierarchy using nvms for hevc motion estimation," in *2019 26th IEEE Int. Conf. on Electronics, Circuits and Systems (ICECS)*, 2019, pp. 162–165.
- [37] —, "Low-power and memory-aware approximate hardware architecture for fractional motion estimation interpolation on hevc," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1–5.
- [38] V. R. Filho *et al.*, "Standalone rate-distortion fme architecture," in *2020 33rd Symposium on Integrated Circuits and Systems Design (SBCCI)*, 2020, pp. 1–6.
- [39] R. Domanski *et al.*, "High-throughput multifilter interpolation architecture for av1 motion compensation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 5, pp. 883–887, 2019.
- [40] D. Freitas *et al.*, "Hardware architecture for the regular interpolation filter of the av1 video coding standard," in *2020 28th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 560–564.
- [41] H. Azgin *et al.*, "A reconfigurable fractional interpolation hardware for vvc motion compensation," in *2018 21st Euromicro Conference on Digital System Design (DSD)*, 2018, pp. 99–103.
- [42] H. Azgin, E. Kalali, and I. Hamzaoglu, "An approximate versatile video coding fractional interpolation hardware," in *2020 IEEE International Conference on Consumer Electronics (ICCE)*, 2020, pp. 1–4.
- [43] D. Palomino *et al.*, "A memory aware and multiplierless vlsi architecture for the complete intra prediction of the hevc emerging standard," in *2012 19th IEEE International Conference on Image Processing*, 2012, pp. 201–204.
- [44] M. Corrêa *et al.*, "High-throughput hevc intrapicture prediction hardware design targeting uhd 8k videos," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017, pp. 1–4.
- [45] G. Pastuszak and A. Abramowski, "Algorithm and architecture design of the h.265/hevc intra encoder," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 210–222, 2016.
- [46] L. Neto *et al.*, "Directional intra frame prediction architecture with edge filter and upsampling for av1 video coding," in *33rd Symposium on Integrated Circuits and Systems Design (SBCCI)*, 2020, pp. 1–6.
- [47] M. M. Corrêa *et al.*, "A high-throughput hardware architecture for av1 non-directional intra modes," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 5, pp. 1481–1494, 2020.

- [48] M. Corrêa *et al.*, “High throughput hardware design for av1 paeth and smooth intra modes,” in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019, pp. 1–5.
- [49] F. Brand, J. Seiler, and A. Kaup, “Intra frame prediction for video coding using a conditional autoencoder approach,” in *2019 Picture Coding Symposium (PCS)*, 2019, pp. 1–5.
- [50] M. Saldanha *et al.*, “Complexity analysis of vvc intra coding,” in *2020 IEEE Int. Conf. on Image Processing (ICIP)*, 2020, pp. 3119–3123.
- [51] H. Azgin, E. Kalali, and I. Hamzaoglu, “An efficient fpga implementation of versatile video coding intra prediction,” in *2019 22nd Euromicro Conference on Digital System Design (DSD)*, 2019, pp. 194–199.
- [52] A. Tissier *et al.*, “Complexity reduction opportunities in the future vvc intra encoder,” in *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, 2019, pp. 1–6.
- [53] X. Zhao *et al.*, “Enhanced multiple transform for video coding,” in *2016 Data Compression Conference (DCC)*. IEEE, 2016, pp. 73–82.
- [54] M. Zheng *et al.*, “A reconfigurable architecture for discrete cosine transform in video coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 3, pp. 810–821, 2019.
- [55] L. Braatz *et al.*, “A new hardware friendly 2d-dct hevc compliant algorithm and its high throughput and low power hardware design,” in *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2019, pp. 654–657.
- [56] R. Calusdian and A. Stillmaker, “Hardware implementation of hevc inverse transform in 45nm cmos,” in *2020 IEEE 11th Latin American Symposium on Circuits Systems (LASCAS)*, 2020, pp. 1–4.
- [57] L. Braatz *et al.*, “A multiplierless parallel hevc quantization hardware for real-time uhd 8k video coding,” in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017, pp. 1–4.
- [58] —, “High-throughput and low-power integrated direct/inverse hevc quantization hardware design,” in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018, pp. 1–5.
- [59] X. Zhao *et al.*, “Nsst: Non-separable secondary transforms for next generation video coding,” in *2016 Picture Coding Symposium (PCS)*, 2016, pp. 1–5.
- [60] A. Kammoun *et al.*, “Forward-inverse 2d hardware implementation of approximate transform core for the vvc standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 4340–4354, 2019.
- [61] I. Farhat *et al.*, “Lightweight hardware implementation of vvc transform block for asic decoder,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 1663–1667.
- [62] Y. Fan *et al.*, “A pipelined 2d transform architecture supporting mixed block sizes for the vvc standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 9, pp. 3289–3295, 2020.
- [63] A. Kammoun *et al.*, “Hardware acceleration of approximate transform module for the versatile video coding standard,” in *2019 27th European Signal Processing Conf. (EUSIPCO)*. IEEE, 2019, pp. 1–5.
- [64] C.-W. Chang, H.-F. Hsu, and C.-P. Fan, “Unified forward and inverse integer transforms design with fast algorithm based hardware sharing architecture,” in *2020 2nd International Conference on Computer Communication and the Internet (ICCCI)*. IEEE, 2020, pp. 126–135.
- [65] F. L. Ramos *et al.*, “Residual syntax elements analysis and design targeting high-throughput hevc cabac,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 2, pp. 475–488, 2019.
- [66] D. Kim, J. Moon, and S. Lee, “Hardware implementation of hevc cabac encoder,” in *2015 International SoC Design Conference (ISOCC)*. IEEE, 2015, pp. 183–184.
- [67] D. Zhou *et al.*, “Ultra-high-throughput vlsi architecture of h. 265/hevc cabac encoder for uhdtv applications,” *IEEE Transactions on circuits and systems for video technology*, vol. 25, no. 3, pp. 497–507, 2014.
- [68] A. V. P. Saggiolato *et al.*, “Hevc residual syntax elements generation architecture for high-throughput cabac design,” in *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, 2018, pp. 193–196.
- [69] H. Schwarz *et al.*, “Quantization and entropy coding in the versatile video coding (vvc) standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.
- [70] C. M. Diniz *et al.*, “A deblocking filter hardware architecture for the high efficiency video coding standard,” in *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2015, pp. 1509–1514.
- [71] F. Rediess *et al.*, “Sample adaptive offset filter hardware design for HEVC encoder,” in *2014 IEEE Visual Communications and Image Processing Conference*, Dec 2014, pp. 299–302, DOI: 10.1109/VCIP.2014.7051563.
- [72] E. Zummach *et al.*, “An uhd 4k@60fps deblocking filter hardware targeting the av1 decoder,” in *2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2020, pp. 1–4.
- [73] S. Midtskogen and J. Valin, “The AV1 Constrained Directional Enhancement Filter (Cdef),” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 1193–1197, DOI: 10.1109/ICASSP.2018.8462021.
- [74] E. Zummach *et al.*, “Efficient hardware design for the av1 cdef filter targeting 4k uhd videos,” in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1–5.
- [75] E. Zummach *et al.*, “High-throughput cdef architecture for the av1 decoder targeting 4k@ 60fps videos,” in *IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS)*. IEEE, 2020, pp. 1–4.
- [76] M. Karczewicz *et al.*, “Vvc in-loop filters,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.
- [77] A. Cerveira *et al.*, “Memory assessment of versatile video coding,” in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 1186–1190.
- [78] A. Cerveira *et al.*, “Memory profiling of h. 266 versatile video coding standard,” in *2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, 2020, pp. 1–4.
- [79] F. Sampaio *et al.*, “Hybrid scratchpad video memory architecture for energy-efficient parallel hevc,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 10, pp. 3046–3060, 2019.
- [80] —, “dsvm: Energy-efficient distributed scratchpad video memory architecture for the next-generation high efficiency video coding,” in *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2014, pp. 1–6.
- [81] —, “Energy-efficient architecture for advanced video memory,” in *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2014, pp. 132–139.
- [82] —, “Approximation-aware multi-level cells stt-ram cache architecture,” in *2015 International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES)*, 2015, pp. 79–88.
- [83] B. Abdallah *et al.*, “Low-complexity transform algorithm for versatile video coding,” in *2019 IEEE International Conference on Design Test of Integrated Micro Nano-Systems (DTS)*, 2019, pp. 1–3.