A Hardware Architecture for Smoothing Filters in H.266/VVC Intra-Prediction

Sara Vitória Henssler
sarahenssler.aluno@unipampa.edu.br
Universidade Federal do Pampa (UNIPAMPA)
Bagé, Brasil

Luciano Volcan Agostini
agostini@ufpel.edu.br
Universidade Federal de Pelotas (UFPel)
Pelotas, Brasil

Marcel Moscarelli Corrêa
marcelcorrea@ifsul.edu.br
Instituto Federal Sul-rio-grandense (IFSul)
Pelotas, Brasil

ABSTRACT

This paper presents a hardware architecture for the preprocessing smoothing filters used in the intra prediction module of the H.266 Versatile Video Coding (H.266/VVC) standard, applicable to both the encoder and decoder sides. The design was fully described in VHDL and synthesized to an Intel Cyclone V FPGA, utilizing 6,961 adaptive logic modules and 4,158 registers, demonstrating feasibility for FPGA implementation with room for integration with a larger intra prediction module. Operating at a maximum frequency of 96.3 MHz without pipelining, the architecture achieves a decoding throughput of up to 2,972 frames per second (fps) for the 1080p resolution, 743 fps for UHD 4K, and 185 fps for UHD 8K, showing that a much lower clock operation can be used for real-time performance. For encoding, the throughput was estimated by targeting a 60 fps rate at 1080p resolution, resulting in an upper bound of approximately 3,171 CUs evaluated per CTU, compatible with fast decision algorithms used in practical encoders.

KEYWORDS

hardware, intra prediction, video coding, h.266, vvc

1 INTRODUCTION

Today, people increasingly rely on digital video services for their leisure, study, work, and communication routines. The encoding of video content for compression purposes becomes indispensable, considering that the amount of data required for the representation of visual content, without compression, becomes impractical. For instance, without compression, an episode of a series in Ultra-High Definition (UHD) 4K resolution (3840x2160 pixels) at 24 frames per second (fps), with a duration of 40 minutes and 24 bits per pixel, has a size of approximately 1.5 terabytes. This exceeds the total storage capacity commonly found in current personal computers, and furthermore, streaming this video in real-time far exceeds the bandwidths commonly available to consumers. However, there are dedicated tools to reduce the amount of data needed for video content, making it storage and streaming feasible tasks. A codec (encoder/decoder) is dedicated to video compression and decompression and can be implemented in software or hardware. In real-time processing scenarios, content must be captured, compressed, and transmitted in extremely short timeframes. Dedicated hardware architectures are better suited for this, as they perform all these tasks with greater speed and lower energy consumption.

The Versatile Video Coding (VVC), also known as ITU-T Rec. H.266 standard [7] was collaboratively developed by the Joint Video Experts Team (JVET) of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). Introduced in 2020 as the successor to H.265 High-Efficiency Video Coding (H.265/HEVC) [6, 10], it brought a significant improvement in compression efficiency, achieving approximately 50% better compression than its predecessor and approximately 75% better than the widely supported H.264/AVC standard [4, 8]. Beyond that, VVC was designed to better meet the needs of diverse digital media applications, including high-resolution formats such as UHD 4K and 8K video, immersive technologies like 360-degree video and virtual reality, as well as high-dynamic range (HDR) content. Its flexibility and efficiency make it particularly suitable for modern streaming platforms, broadcasting, video conferencing, and emerging interactive media.

VVC is still very recent, so it will take time before dedicated hardware architectures with support for it become widely available. So far, the Intel Lunar Lake CPU architecture [5] offers support for VVC decoding, as well as the Hantro VC9000D Intellectual Property (IP) core hardware decoder from Verisilicon [11]. As for VVC hardware accelerated encoding, the Allegro IP Core E320 Encoder [1] seems to be the only one available.

The closest related works are [2][3] from Borges et al., where the authors present two low-power and high-performance hardware designs for the angular intra prediction of VVC. However, despite being a mandatory step, the preprocessing smoothing filters are not mentioned in these works.

Therefore, this paper presents a hardware design for the preprocessing smoothing filters used in intra prediction of VVC. The architecture was developed in VHSIC Hardware Description Language (VHDL) and synthesized using Quartus Prime, allowing its implementation on FPGA platforms and validating its practical feasibility.

This paper is organized as follows: Section 2 provides a brief background on intra prediction in the VVC standard. Section 3 describes the proposed architecture and the design process, while Section 4 presents and discusses the results obtained from the proposed architecture. Finally, Section 5 concludes this paper.

2 INTRA PREDICTION IN VVC

Video frames may contain a large amount of redundancy, which can be explored and reduced by the encoder. The intra-frame prediction process in VVC reduces the spatial redundancy present within a

In: Proceedings of the Brazilian Symposium on Multimedia and the Web (WebMedia'2025). Rio de Janeiro, Brazil. Porto Alegre: Brazilian Computer Society, 2025.

ISSN 2966-2753

frame, based on references contained in the same frame. In VVC, this tool has been refined and expanded compared to its predecessor, HEVC, both in terms of the variety of modes and structural flexibility, allowing better adaptation to different texture patterns, higher resolutions, and greater visual complexity. Further details are described below.

2.1 Hierarchical Block Division

In VVC, the frame is partitioned into blocks called Coding Tree Units (CTUs), with a fixed size of 128x128 pixels. VVC inherits from HEVC the Quadtree (QT) partitioning structure and extends this approach by introducing an additional structure called Multi-Type Tree (MTT), resulting in the partitioning structure known as Quadtree with Nested Multi-Type Tree (QTMT). In this scheme, the CTU is initially subdivided using the QT, which allows only quaternary splits into four parts. Each leaf resulting from the QT tree can then be further subdivided by the MTT structure, which provides greater flexibility. As illustrated in Figure 1, the QTMT structure allows six different types of partitioning, in which each block, called a Coding Unit (CU) in VVC, can be subdivided quaternarily, binarily, or ternarily. Each leaf of the QT can be split horizontally or vertically into two or three parts. This subdivision process can be repeated until the smallest allowed size for a CU is reached, allowing for a wide range of sizes, such as: 4x4, 4x8, 8x4, 8x8, 8x16, 16x8, 16x16, 16x32, 32x16, 32x32, 32x64, 64x32, 64x64, 64x128, 128x64, and 128x128 pixels [7].

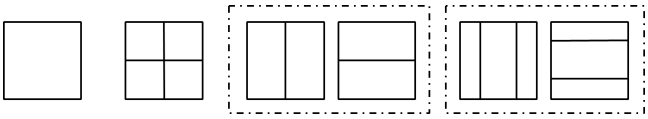

Figure 1: H.266/VVC block partitioning scheme.

2.2 Prediction Modes

VVC defines a total of 67 main modes for intra prediction: Planar, DC, 65 angular modes. The Planar mode produces a smooth gradient prediction by interpolating pixel values from the edges of the block, resulting in a gentle transition between neighboring pixels, while the DC mode predicts all pixels in the block with a single constant value, which is the average of the reference samples around the block, capturing flat or uniform areas efficiently. Lastly, Angular modes predict pixel values inside a block by propagating reference samples along specific directional angles, where each mode corresponds to a different angle or direction (e.g., horizontal, vertical, diagonal, and many intermediate angles) [7].

2.3 Pre and Postprocessing Techniques

Several techniques are applied in VVC during the intra prediction process to improve prediction performance. In preprocessing, tools like reference sample smoothing filters and, when necessary, Intra Sub-Partitions (ISP) are applied. ISP consists of vertical or horizontal subdivisions within the block, which are predicted separately. In postprocessing, techniques such as Residual Differential Pulse Code Modulation (RDPCM) and Cross Component Linear Mode (CCLM) are used to improve the prediction residue or to apply a simple prediction in order to enhance encoding [7].

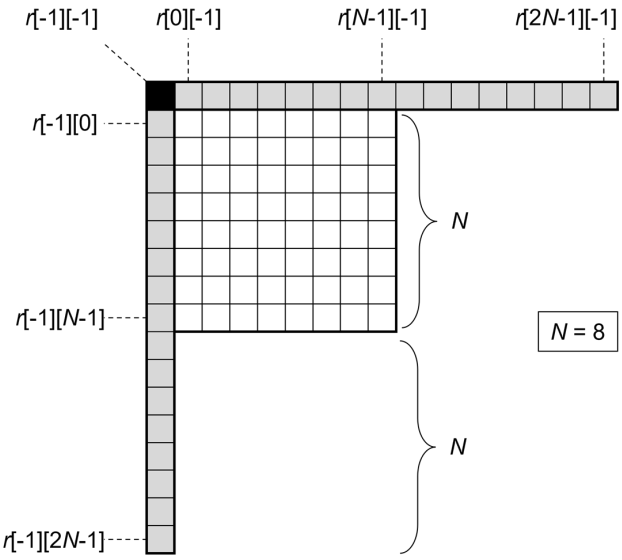

Figure 2: Samples array r for a 8x8 CU in VVC intra prediction. White squares are samples to be predicted and non-white are reference samples.

2.4 Reference Samples Filters

Among these steps cited in Section 2.3, the one of interest for this work is the application of smoothing filters to reference samples, performed during preprocessing. Before using the reference samples for prediction purposes, VVC can apply smoothing filters, to remove noise or abrupt transitions that could hinder the prediction process. To better illustrate this process, Figure 2 presents an example of a block to be predicted (in white). The block is accompanied by two arrays of reference samples, column and row, located on the top and left edges (in gray), which contain twice the number of samples relative to each dimension of the block, in addition to including the sample at index [-1][-1] (in black), common to both the column and row of reference samples.

Equations 1, 2 and 4 describe a linear 1D smoothing filter (3-tap filter) with weights {1, 2, 1}, with the adjacent reference samples to the one being filtered being assigned a weight of 1, while the sample itself receives a weight of 2. Equation 1 describes a special case, because the sample at index [-1][-1] needs one sample from the column of reference samples and another from the row. Lastly, Equations 3 and 5 are also special cases, because the last sample in each array are not filtered. In all equations, the resulting value p[x][y] replace the reference r[x][y].

$$p[-1][-1] = (r[-1][0] + 2r[-1][-1] + r[0][-1] + 2) \gg 2 \quad (1)$$

$$p[-1][y] = (r[-1][y+1] + 2r[-1][y] + r[-1][y-1] + 2) \gg 2 \quad (2)$$

$$p[-1][H-1] = r[-1][H-1] \tag{3}$$

$$p[x][-1] = (r[x-1][-1] + 2r[x][-1] + r[x+1][-1] + 2) \gg 2 \tag{4}$$

$$p[W-1][-1] = r[W-1][-1] \tag{5}$$

It is important to note that filtering is not always applied. The encoder must check certain conditions on the fly, such as the size of the block to be predicted, the availability of neighboring blocks, and the prediction modes used in the previously encoded reference blocks, among others.

3 PREPROCESSING FILTERS DESIGN

The proposed architecture consists of a preprocessing filter for variable block sizes to be used in VCC intra prediction. These filters follow the ITU-T Recommendation H.266 [7]. Figure 3 presents a top-level view of the architecture. It receives a row and a column of reference samples of variable sizes, including the top-left sample common to both arrays. Additionally, it takes in control signals, which include the size of the block (current CU being predicted by the encoder), the availability of the reference samples, the intra mode being tested with the block, and the intra modes used to encode the two neighboring blocks. More detailed information about each block and about the flow of data can be found below.

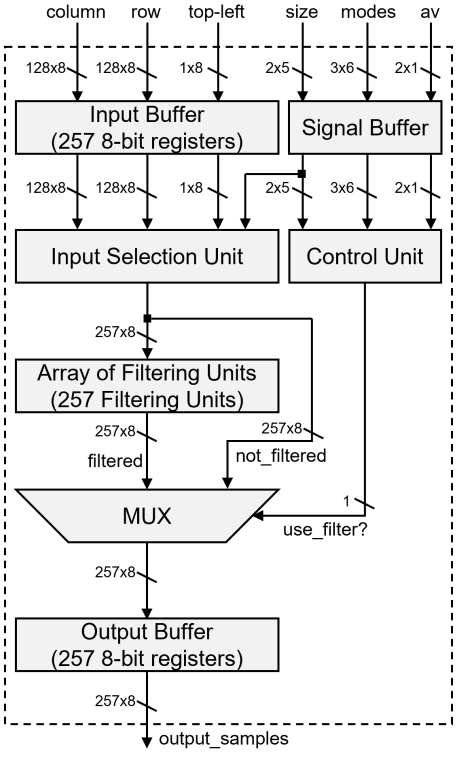

Figure 3: Block diagram of the top-level architecture.

3.1 Buffers

The architecture includes a buffer to store the input reference samples, which consist of 128 samples from the column to the left of the block, 128 samples from the row above the block, and one additional sample common to the two reference arrays. These samples are illustrated in Figure 2 in gray and black. All reference samples are 8-bit luminance values. Additionally, a second buffer of equal size is used to store the outputs of the smoothing filter process, and a small buffer is used for control signals.

In total, the design requires 514 8-bit registers for storing input and output data, three 6-bit registers for prediction mode signals, two 5-bit registers for block size information, and two 1-bit registers for availability information.

3.2 Input Selection Unit

Since the architecture processes reference samples for the largest block size in parallel, a selection unit is required to handle smaller blocks efficiently. This unit implements Operand Gating, selecting only the valid samples based on the current block size. It receives both the reference arrays from the input buffer and their size from the signal buffer, then filters the data: valid entries are passed through unchanged, while unused entries are zeroed out. This reduces unnecessary switching activity in the data path that follows, thereby lowering dynamic power consumption of the circuit.

3.3 Control Unit

This unit is responsible for deciding whether the output of the smoothing filter process will consist of the original reference arrays (unchanged) or the filtered ones. It receives the block size information, the current prediction mode to be tested, the prediction mode of its two neighboring blocks and two flags signaling the availability of these neighboring samples. Based on this data, the unit tests whether the requirements cited in Section 2 of this paper are met and outputs a single 1-bit signal that activates or disables the whole filtering process through an array of multiplexers.

3.4 Array of Filtering Units

The filtering unit is the smallest operative unit in the architecture and consists of the 3-tap filter responsible for preprocessing the samples according to equations 1, 2 and 4. Figure 4 illustrates the structure of this unit. In the figure, ref_A and ref_C are the adjacent samples, while ref_B is the one being filtered.

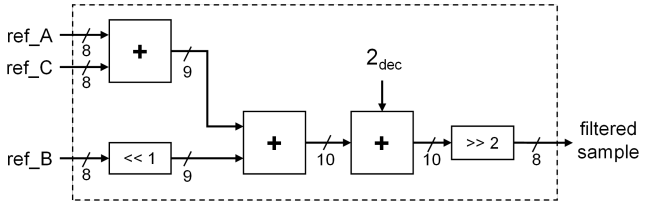

Figure 4: Filtering unit block diagram

The array of filtering units replicates 127 filtering units to process a column of reference samples, 127 more for a row of references, and a single extra filtering unit for the sample at index [-1][-1]. The

output of the array of filtering units is forwarded to the control multiplexer.

Although the smoothing filter is applied identically in both encoding and decoding, the chosen level of parallelism was designed with the video encoder in mind, because the prediction module of the encoder, which includes the smoothing filter, demands significantly higher throughput.

4 SYNTHESIS RESULTS

To confirm the feasibility of the proposed architecture in the early stage of development, the design was fully described in VHDL and synthesized using Intel Quartus Prime Lite Edition (version 23.1std1). The target platform was an FPGA from the Cyclone V mid-range family, specifically the 5CGXFC7C7F23C8 device. The synthesis results were obtained under the "Slow 1100mV 85°C" timing model, representing a worst-case scenario for timing analysis. Under these conditions, the design achieved a maximum operating frequency of 96.3 MHz. Regarding resource utilization, the architecture required:

- 6,961 Adaptive Logic Modules (ALMs), representing 12.33% of the total (56,480) used as combinational logic elements in the design;
- 4,158 1-bit registers, representing 1.84% of the total (225,920) required to implement the sequential logic required for integration with a larger intra prediction module.

These results demonstrate that the proposed architecture is viable for FPGA implementation with low resource usage. The logic and register utilization remain within practical limits for medium-sized FPGAs, leaving room for integration with other functional blocks required by the intra prediction process.

For a throughput analysis, unlike the encoder, the decoder does not explore multiple block partitioning options, it simply applies the operations signaled in the bitstream. Therefore, we can assume a continuous and deterministic workload. To this end, we assumed a worst-case scenario where all blocks in a bitstream are predicted as 8x8 CUs, which is smallest block size to undergo the smoothing filter, hence resulting in the highest amount of blocks passing through the smoothing filter. Since the proposed design is able to process the filtering for one input block of any size per cycle, the Expressions 6 to 8 give the decoding throughput for the 1080p, UHD 4k and UHD 8K resolutions, respectively.

$$1080\text{p} : \frac{96{,}300{,}000 \text{ blocks/sec}}{32{,}400 \text{ blocks/frame}} \approx 2{,}972 \text{ frames/sec} \quad (6)$$

$$\text{UHD 4K} : \frac{96{,}300{,}000 \text{ blocks/sec}}{129{,}600 \text{ blocks/frame}} \approx 743 \text{ frames/sec} \quad (7)$$

$$\text{UHD 8K} : \frac{96{,}300{,}000 \text{ blocks/sec}}{518{,}400 \text{ blocks/frame}} \approx 185 \text{ frames/sec} \quad (8)$$

On the other hand, estimating the maximum encoding throughput is a non-trivial task due to flexibility of block division in VVC. In other words, the Rate-Distortion Optimization (RDO) [9] mode decision pipeline introduces a highly data-dependent flow where a CTU can be recursively split into a large number of CUs of varying sizes and shapes, especially when no early termination or pruning strategies are employed. In such cases, the number of candidate blocks evaluated for a single CTU can grow exponentially, approaching a combinatorial explosion, which is why not even the reference software explores all the possibilities. This makes it extremely difficult to define a reliable worst-case throughput scenario without overestimating the computational burden. A realistic performance evaluation would have to be tied to an existing encoder implementation and use its decision heuristics as reference.

Therefore, to estimate the encoding throughput, a top down approach based on a real time target was adopted, rather than attempting to model the combinatorial complexity of VVC block partitioning. Specifically, a 1080p resolution at 60 fps target was set, which equals approximately 506 CTUs per frame and a required throughput of 30,360 CTUs per second. Given that the proposed architecture operates at a maximum frequency of 96.3 MHz and is capable of processing one block per clock cycle, up to 96.3 million blocks per second can be processed. Dividing this by the number of CTUs per second required results in approximately 3,171 candidate blocks per CTU, as shown by Expression 9. This represents the maximum number of block evaluations that can be supported in real time encoding of 1080p videos at 60 fps, which means that an encoder using the smoothing filter design proposed in this paper has to operate within this limit of block partitioning per CTU to achieve real time performance. This approach provides a practical and encoder aware throughput ceiling that can guide integration with real world encoding pipelines.

$$1080\text{p@60fps} : \frac{96{,}300{,}000 \text{ blocks/sec}}{30{,}360 \text{ CTUs/sec}} \approx 3{,}171 \text{ blocks/CTU} \quad (9)$$

5 CONCLUSION

This paper presented a hardware design for the preprocessing smoothing filters used in the VVC intra prediction, for both the decoder and encoder sides. The design was verified in an Intel Cyclone V FPGA device and used 6,961 ALMs and 4,158 registers, which showed that the proposed architecture is viable for FPGA implementation with room for integration with an intra prediction module. The proposed design achieved a maximum operating frequency of 96.3 MHz, without pipelining to shorten the critical path. Owing to its high parallelism, this enables decoding throughputs of 2,972, 743, and 185 fps for 1080p, UHD 4K, and UHD 8K resolutions, respectively. Thus, real-time decoding can be achieved at significantly lower frequencies. For encoding, a different estimation was used: targeting 1080p @ 60 fps, the encoder can evaluate up to 3,171 CUs per CTU during block partitioning—which is reasonable considering fast mode decision algorithms found in most software codecs.

Future work includes synthesizing the architecture with standard-cell ASIC libraries to better assess area, timing, and especially power consumption, which cannot be reliably measured on FPGA. These results will help evaluate the design's efficiency and its suitability for power-constrained environments such as mobile or embedded encoders and decoders.

ACKNOWLEDGMENTS

This research was supported by the Instituto Federal Sul-rio grandense (IFSul) and Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS).

REFERENCES

[1] Allegro DVT. 2024. *E320 VVC Encoder Video IP.* https://www.allegrodvt.com/products/e320-vvc-encoder-video-ip/ [Online; accessed July 2025].

[2] V. Borges, M. Perleberg, M. Porto, and L. Agostini. 2023. Efficient Architecture for VVC Angular Intra Prediction based on a Hardware-Friendly Heuristic. In *2023 IEEE 14th Latin America Symposium on Circuits and Systems (LASCAS)*. Quito, Ecuador, 1–4. https://doi.org/10.1109/LASCAS56464.2023.10108393

[3] V. Borges, M. Perleberg, M. Porto, and L. Agostini. 2024. High-Throughput Hardware Design for the Complete VVC Angular Intra Prediction. In *2024 31st IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. Nancy, France, 1–5. https://doi.org/10.1109/ICECS61496.2024.10848892

[4] Benjamin Bross, Jianle Chen, Jens-Rainer Ohm, Gary J. Sullivan, and Ye-Kui Wang. 2021. Developments in International Video Coding Standardization After AVC, With an Overview of Versatile Video Coding (VVC). *Proc. IEEE* 109, 9 (2021), 1463–1493. https://doi.org/10.1109/JPROC.2020.3043399

[5] Intel Corporation. 2024. *Intel® Core™ Ultra Processors Series 2.* https://www.intel.com/content/www/us/en/ark/products/series/241071/intel-core-ultra-processors-series-2.html#@Mobile [Online; accessed July 2025].

[6] ITU-T Recommendation. 2013. High Efficiency Video Coding. https://www.itu.int/rec/T-REC-H.265. ITU-T Recommendation H.265.

[7] ITU-T Recommendation. 2020. Versatile Video Coding. https://www.itu.int/rec/T-REC-H.266. ITU-T Recommendation H.266.

[8] Benjamin Bross; Ye-Kui Wang; Ye-Kui Ye; Shan Liu; Gary J. Sullivan. 2021. Overview of the Versatile Video Coding (VVC) Standard and Its Applications. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 10 (Oct. 2021), 3736–3764. https://doi.org/10.1109/TCSVT.2021.3101953

[9] G.J. Sullivan and T. Wiegand. 1998. Rate-distortion optimization for video compression. *IEEE Signal Processing Magazine* 15, 6 (1998), 74–90. https://doi.org/10.1109/79.733497

[10] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. 2012. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology* 22, 12 (2012), 1649–1668. https://doi.org/10.1109/TCSVT.2012.2221191

[11] VeriSilicon Holdings Co., Ltd. 2024. *VeriSilicon Unveils Hantro VC9000D Multi-Format Video Decoder.* https://verisilicon.com/en/PressRelease/HantroVC9000D [Online; accessed July 2025].