MEMORY-AWARE, LOW-POWER AND HIGH-THROUGHPUT AV1 FME INTERPOLATION ARCHITECTURE

WILLIAM KOLODZIEJSKI[1]; ROBSON DOMANSKI[2]; LUCIANO AGOSTINI[3]

[1]UFPel – wkolodziejski@inf.ufpel.edu.br
[2]UFPel – radomanski@inf.ufpel.edu.br
[3]UFPel – agostini@inf.ufpel.edu.br

1. INTRODUCTION

The advent of the COVID-19 pandemic has pushed up even more the use of digital videos, which was already in great increase over the last years. One clue of this assumption is the decision of Netflix, Amazon Prime, and YouTube to reduce the video quality to guarantee the quality of service (T. I. EXPRESS, 2020). Also, Statista (2020) estimated that by 2022, digital video content would be responsible for the traffic of about 77.49 EiB/month (1 EiB is 2^{60} bytes), meaning approximately 82% (CISCO, 2018) of the global Internet traffic, which may be underestimated since no pandemic was in sight in that moment. Under this scenario, video compression is being used like it never was before, and this implies in the necessity to improve the current video encoders, in order to reduce the power dissipation and time consumed during the video compression process. One of these video encoders is the AV1, released in 2018 by Alliance for Open Media.

One of the most used AV1 tools is the Fractional Motion Estimation (FME), responsible to generate interpolated samples, using integer samples fetched from the memory. To perform the interpolations, the AV1 FME uses a total of 90 Finite Impulse Response FIR filters (HAN, 2020), organized in six families of 15 filters each. Among the current video codecs, the AV1 uses the largest number of FIR filters in the FME interpolation process. The VVC, for example, uses 78 filters (JVET, 2021). The AV1 FME filters have from 2 up to 8-taps, achieving an accuracy of 1/8 and 1/16 samples, for luminance and chrominance, respectively, allowing for half, quarter, eighth, and sixteenth precision (HAN, 2020).

This work presents a dedicated memory-aware, low-power and high-throughput hardware design for the AV1 FME interpolation able to process up to UHD 8K@30fps, supporting all filters and block sizes. This work is a continuation of a previous work (DOMANSKI, 2020). The method used to develop this architecture consists on the proposal of an optimized combinational multiplier less multi-filter solution, approximating the filters coefficients to hardware-friendly values and reducing the number of taps.

2. METHODOLOGY

In order to simplify the AV1 FME interpolation filters, the first hardware-friendly approximation explored was to transform the required multiplications defined in the filters into shifts or shift-adds operations, using an approximation of the original taps. The proposed approximation considered the fact that the two central taps are the most important ones to the interpolation result and defined their values as two shift-adds, improving the precision. On the other hand, since the most peripheral taps are less important, their values were defined using only one shift. Another approximation proposed is to reduce the number of taps to a maximum of four in all FME filters, mainly to reduce the memory bandwidth.

Figure 1 presents the proposed architecture, called Basic Multiplier less Multi-filter (BMM). This set of shifters, adders and multiplexers is able to compute all the 90 AV1 approximated FME interpolation filters (one at a time). Each *A* in Figure 1 is an input sample to be multiplied by a filter tap. Since the maximum number of supported taps in this architecture is four, there are four inputs. The shifts and adds are responsible to generate each approximate filtered value and the multiplexes are responsible to select which shifter should be used for each filter. The first BMM operation over the inputs is a shift-left of *n* bits, which is equivalent to a multiplication by 2^n.

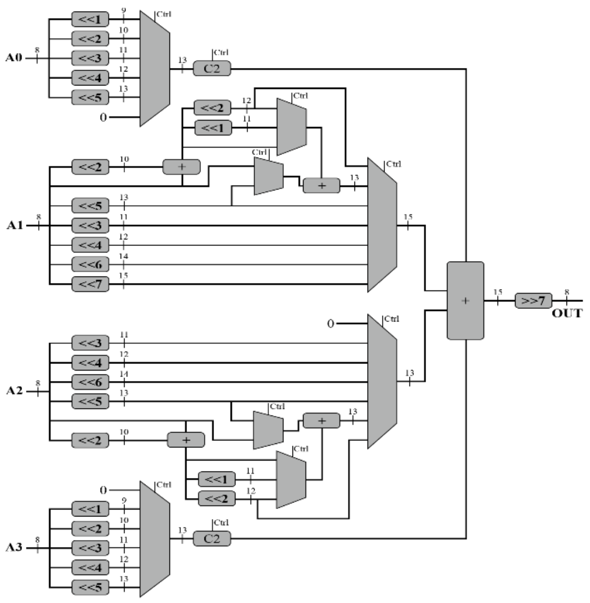
Figure 1. Basic Multiplier-less Multi-filter (BMM).

The output of multiplexers connected to *A0* and *A3* inputs have a controlled complement-of-two operation (C2 in Figure 1) to correct the results signal. The signal inversion is used to generate negative taps and it is not applied if a positive tap is required. Then, all outputs are added, and the last step is a 7-bit shift-right to divide by 128 the previous result, generating the interpolated sample.

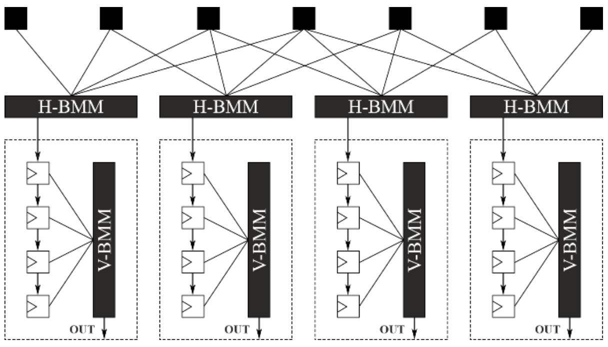

Figure 2. AM4AF: Approximate FME Interpolation Architecture.

Figure 2 presents the Approximate Multiplier less 4-tap-Approximate Filter (AM4AF) architecture. Both H-BMM and V-BMM are the same filter presented in Figure 1, where the difference in their names is only to highlight either is a horizontal (H-BMM) or vertical (V-BMM) instance. Then, the instances are connected to through a shift-register-chain of 4 positions. These positions correspond to the 4 BMM inputs (*A*s in Figure 1).

The AM4AF was designed to process 4x4 blocks, which is the smallest block size supported by AV1. Since any bigger block can be build up from 4x4 sub blocks, the architecture supports all AV1 block sizes, ranging from 4x4 to 128x128.

The precise AV1 interpolation requires a matrix with 11x11 samples to interpolate a 4x4 block. Since four taps were removed in this architecture, 4 rows and 4 columns can also be removed from this matrix, implying in a 7x7 matrix. This reduction leads to a reduction in data fetched from the memory in 59.5%.

The architecture was designed to read one line of this 7x7 matrix at each clock cycle and, then, shift the registers until filling the chain (these shifts correspond to the combinational distribution of the sample inputs). The AM4AF takes four clock cycles to start processing the V-BMM instances and three clocks to finish the processing, taking seven clocks to generate an interpolated 4x4 block.

One observation is that both the memory usage and memory bandwidth are highly dependent of the whole AV1 hardware implementation. Independently of the memory technology, the 59.5% reduction is related to the FME interpolation process, when compared to a standard-defined precise FME.

3. RESULTS AND DISCUSSION

The proposed approximations were evaluated in comparison to the original filters using 9 video sequences from Mozilla and Netflix data set (N. W. GROUP, 2020). The evaluation considered four CQs = {20, 32, 43, 55} enabling all AV1 encoding tools. CQ stands for Constant Quality and ranges from 0 (no quality loss) to 63 (maximum quality loss). The experiments were done using the libaom, the AV1 reference software, version 2.0.0 (AOMedia, 2020). The coding efficiency evaluation was done using the Bjontegaard Delta Bit Rate (BD-BR) metric (BJONTEGAARD, 2008). This metric indicates, for the same objective video quality, the percentage difference of bit-rate required to represent a video. The closest to zero, the more similar to the original video. Negative values mean that less bits were required to represent the same result. The experiments were done following the Video Codec Testing and Quality Measurement (N. W. GROUP, 2020).

Table 1 presents the results of coding efficiency of the proposed approximations in comparison to the original filters, grouping the videos by resolution. The first observation is that the higher the video resolution, the lower tends to be the approximation impact on BD-BR. In some cases (negative values) the approximations can even increase the coding efficiency.

Sequence	Resolution	BD-BR (%)
Arena of Valor	Full HD	0.67
Market Place		2.46
Square and Time-lapse		2.62
Tunnel Flag		5.21
Foreman	UHD 4K	0.82
Coastguard		-0.14
Cactus		1.28
Bar Scene		-1.90
Boxing		2.06
Average	**Overall**	**1.58**

Table 1. Coding Efficiency Results

The proposed approximations caused a small degradation in coding efficiency and the results are as better as higher is the video resolution. When compared to our previous work (DOMANSKI, 2021), which reached an average BD-BR of 0.54%, the novel overall BD-BR results are slightly higher, but with important improvements in memory bandwidth.

Three versions of the AVI FME interpolation were synthesized to allow comparisons: the approximate architecture, a precise version without simplifications and the previous design of our group, proposed by Domanski (2021). The precise version uses multipliers instead of shifters. All architectures follow the same architectural template (presented in Figure 2) and were described in VHDL and synthesized for a 40nm TSMC standard-cells technology with 1.1V using Cadence RTL Compiler tool. The power results were generated using the default tool switching activity (20%). The gate count was calculated based on 2-input NANDs size (0.9408mm^2). The frequency was defined as 686MHz since this is the required frequency to process UHD 8K@30fps for the precise version.

Table 2 presents the synthesis results along with a comparison with the other three architectures. The comparison with the precise version showed that our architecture uses 84.6% less area and dissipates 83.4% less power than the precise version. When compared to Domanski (2021), the gains were of 41.8% in area and 7.1% in power. At this point it is important to highlight that the power results considered only the static and dynamic power of the designed hardware and did not consider the power reduction caused by the reduction in the memory bandwidth.

Work	Freq. (MHz)	Gates (K)	Power (mW)
Precise	686.0	275.7	149.7
DOMANSKI	686.0	72.6	26.7
AM4AF	**686.0**	**42.2**	**24.8**

Table 2. Synthesis Results

4. CONCLUSIONS

This work presented an optimized approximate architecture for the AV1 FME interpolation filters that can process up to UHD 8K@30fps, where t The original AV1 FME filter taps were approximated to hardware-friendly values and the maximum number of filter taps was limited to four. This architecture is a memory-aware, low-area, low-power, and high-throughput hardware design exploring approximate computing which supports all 90 AV1 FME interpolation filters. Besides important improvements in area, and power, the proposed approximations allowed a memory bandwidth reduction of 59.5% in comparison to the state-of-the-art solutions. When considering a precise version, our architecture is also able to reduce the area in 84.6% and reach power savings of 83.4% (without consider the power reduction generated by the memory bandwidth reduction), at a cost of 1.58% in BD-BR.

5. BIBLIOGRAPHY

COVID-19 impact: Streaming services to dial down quality as internet speeds fall. T. I. Express, Mar. 2020. Access on Jul. 12, 2021. Online. Available: https://indianexpress.com/article/technology/tech-news-technology/coronavirus-internet-speeds-slow-netflix-hotstar-amazon-prime-youtube-reduce-streaming-quality-6331237

Global mobile data traffic 2017-2022. Statista, Oct. 2019. Access on Jul. 25, 2021. Online. Available: https://www.statista.com/statistics/271405/globalmobiledata-traffic-forecast.

Cisco visual networking index: Forecast and methodology, 2016–2021 – cisco. Cisco, Oct. 2018. Access on Jul. 25, 2021. Online. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html

AV1 codec library. AOMedia, May. 2020. Access on Jul. 26, 2021. Online. Available: https://aomedia.googlesource.com/aom/+/refs/tags/v2.0.0

VVC codec library. JVET, Abr. 2021. Access on Jul. 26, 2021. Online. Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM

Video codec testing and quality measurement. N. W. Group, Aug. 2020. Access on Jul. 26, 2021. Online. Available: https://tools.ietf.org/html/draft-ietf-netvc-testing-09

DOMANSKI, R.; ET AL. Low-Power and High-Throughput Approximated Architecture for AV1 FME Interpolation. **2021 IEEE International Symposium on Circuits and Systems (ISCAS)**, p. 1-5, 2021.

HAN, J.; ET AL. A technical overview of AV1. **Proceedings of the IEEE**, p. 1–28, 2021.

BJONTEGAARD, G. Improvements of the BD-PSNR model. **35th VCEG Meeting**, Berlin, Germany, 2008.