



## PROJETO DE ARQUITETURA DE FILTRO DE DEBLOCAGEM PARA O DECODIFICADOR AV1

EDUARDO ZUMMACH; JONES GOEBEL; ROBERTA PALAU; LUCIANO AGOSTINI; MARCELO PORTO

Universidade Federal de Pelotas – {evzummach, jwgoebel, rcnpalau, **agostini** e **porto**}@inf.ufpel.edu.br

### 1. INTRODUÇÃO

O novo contexto mundial, em que o isolamento social causado pelo COVID-19 modificou o dia a dia das pessoas, evidencia ainda mais a importância da codificação de vídeo. Este processo praticamente invisível ao usuário permite que as pessoas possam ficar próximas apesar do distanciamento social, através de aplicações de *streaming*, tais como chamadas de vídeo e conferências. Estas tecnologias já eram consideradas importantes anteriormente, porém hoje se tornaram essenciais no cotidiano das pessoas.

Como exemplo, desde que o isolamento social começou, o consumo de vídeo e o uso de internet aumentaram, sendo que o Facebook, o Instagram e a Netflix foram obrigados a reduzir a qualidade dos vídeos para poder manter a demanda atual (TECHTUDO, 2020). Isto é, tiveram que comprimir ainda mais os vídeos para atender esta nova demanda e existem tecnologias de codificação altamente eficientes hoje em dia para esse fim. Uma destas tecnologias é o *AOM Video 1* (AV1), novo formato de vídeo, livre de *royalties*, fundado em 2018 pela *Alliance for Open Media* (AOM), um consórcio entre grandes empresas, tais como Google, Microsoft, Intel, Cisco, Netflix, Mozilla e Facebook.

Conforme outros padrões de codificação, o AV1 segue um esquema híbrido de codificação baseado em blocos, composto por predições inter-quadros e intra-quadros, transformadas, quantização, codificação de entropia, quantização inversa, transformada inversa, e, por fim, os filtros de laço, composto por três filtros: (i) o *Deblocking Filter* (DBF), (ii) o *Constrained Directional Enhancement Filter* (CDEF), e (iii) o *Switchable Loop Restoration Filter* (SLRF) (AOM, 2018). Estes filtros tem o objetivo de melhorar a qualidade subjetiva do vídeo.

O filtro de deblocagem (DBF) é o foco deste trabalho, e é utilizado para reduzir ou eliminar artefatos causados pelo processo de codificação, tais artefatos são chamados de efeitos de blocos. Considerando que as operações no processo de codificação de vídeo são baseadas em blocos de amostras, e que estes blocos são processados separadamente dos blocos vizinhos e com o uso de ferramentas distintas, as bordas entre esses blocos após serem processados podem apresentar discontinuidades que acabam degradando a qualidade da imagem. O DBF tem o objetivo de atenuar tais artefatos melhorando a qualidade visual do vídeo. Outros codecs, tal como o *High Efficiency Video Coding* (HEVC) também possuem um filtro de deblocagem. Porém, apesar de o DBF do AV1 e o DBF do HEVC possuírem os mesmos objetivos, seus algoritmos são diferentes. A maior diferença entre eles é o tamanho da borda avaliada entre os blocos adjacentes: no HEVC os quadros são divididos em blocos 8x8 para a aplicação, enquanto que no AV1 os blocos usados são os blocos de transformadas, que variam de tamanhos 4x4 até 64x64 e considerando tamanhos assimétricos (AOM, 2018).

Desta forma, este trabalho tem o objetivo de apresentar um projeto de arquitetura de hardware para o DBF do decodificador do AV1, baseado em operações de filtragem das bordas internas entre os blocos de transformadas adjacentes, com a implementação dos quatro filtros determinados no seu processo.

### 2. METODOLOGIA

Para realizar este trabalho, foi necessário um estudo de artigos científicos, tais como (LEI, 2017) e (CHEN, 2018). Uma análise da documentação (AV1, 2019) e do seu código (AOM, 2018). Uma avaliação do *Codec* VP9 (GRANGE, 2016), no qual o o DBF do AV1 foi baseado com fins de compreender o comportamento do filtro.

A arquitetura de hardware foi descrita em VHDL e validada utilizando a ferramenta ModelSim. Para os resultados apresentados na sessão 4, foram considerados o uso da síntese em ASIC a partir da biblioteca standard cells TSMC 40nm utilizando 0.9 V e 25°C. A síntese foi realizada a partir da ferramenta Cadence Genus Synthesis Solution e os resultados de potência foram estimados a partir do padrão de atividade de chaveamento (20%). A área em *gates* foi calculada baseada no número de 2-input NANDS ( $0.9408 \mu\text{m}^2$ ).

### 3. FILTRO DE DEBLOCAGEM AV1

Conforme mencionado, o DBF foi desenvolvido com o intuito de suavizar as bordas entre blocos adjacentes e, assim, melhorar a qualidade subjetiva do vídeo, reduzindo artefatos associados à semi-independência entre esses blocos.

A Figura 1 apresenta a importância da atuação de um filtro de deblocação. Na sequência é apresentado o quadro original, ou seja, antes da compressão (Figura 1 (a)), a (Figura 1 – (b)) apresenta o quadro já codificado, altamente comprimido, onde é possível observar transições bruscas entre os blocos processados, resultado do processo de codificação. No entanto, na Figura 1 – (c) estas transições são suavizadas pelo filtro de deblocação, dando um melhor aspecto ao quadro, apesar da compressão intensa.



Figura 1 – Atuação do filtro sobre o quadro codificado

O DBF atua nas bordas dos blocos, avaliando primeiramente todas as bordas verticais, e então as horizontais. Existem decisões específicas de filtragem que definem qual filtro será aplicado com base no tamanho do bloco, que pode assumir um tamanho específico dentre 19 tamanhos. Essa escolha pode definir um dos quatro filtros definidos no DBF, conforme demonstrado no fluxograma da Figura 2: um filtro de 13 taps (taps [1,1,1,1,1,2,2,2,1,1,1,1,1]), 7 taps (taps [1,1,1,2,1,1,1]), 5 taps (taps [1,2,2,2,1]) ou um de 4 taps (definidos dinamicamente). Estes filtros são chamados de Filter14, Filter8, Filter6 e Filter4, respectivamente, sendo que o número representa quantas amostras são utilizadas para uso.

O plano de luminância pode utilizar o Filter14, Filter8 ou o Filter4, com base no tamanho: Filter14 é aplicado em bordas maiores que 8 amostras, Filter8 em bordas com 8 amostras e Filter4 nos outros casos. Já o plano de crominância utiliza o Filter6 para bordas maiores que 4 amostras e o Filter4 para os outros casos.

Após definição do filtro com base no tamanho do bloco, esta escolha é reajustada utilizando a relação de variação das amostras no conjunto a ser filtrado. Esta refinação tem o propósito de evitar borrar bordas bem definidas. Neste processo alguns sinais que indicam a escolha são definidos: *filter\_mask*, *flat\_mask\_narrow* (flat1) e *flat\_mask\_wide* (flat2). A *filter\_mask* é definida com base em dois limites utilizados nas inequações:

$$|x_i - x_{i-1}| > Lim_0 \begin{cases} i = 1, f\_len = 4 \\ i \in \{1, 2\}, f\_len = 6 \\ i \in \{1, 2, 3\}, f\_len \geq 8 \end{cases} \quad (1)$$

$$2 * |p_0 - q_0| + \frac{|p_1 - q_1|}{2} > Lim_1 \quad (2)$$

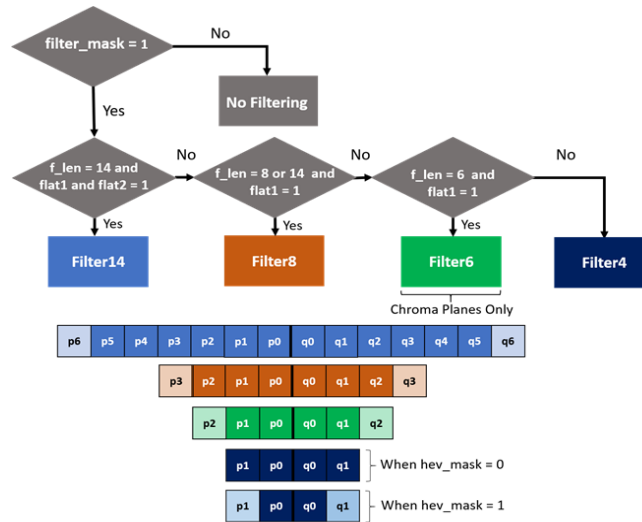


Figura 2 – Fluxograma de aplicação do DBF do AV1

No qual  $x_i$  indica uma amostra  $p$  (amostras à esquerda da borda) ou  $q$  (amostras à direita), na posição  $i$  em relação a borda, conforme o fluxograma. Para o cálculo desta máscara, são necessários os limites  $Lim_0$  e  $Lim_1$ , recebidos do codificador. Caso ambas inequações sejam falsas,  $filter\_mask$  é definida como 1.

Após isto, é calculada a planicidade do conjunto de amostras de cada lado da borda, para evitar outros artefatos de codificação. A partir da inequação abaixo, são definidas as máscaras flat2 e flat1:

$$|x_i - x_0| \leq 1, \begin{cases} i \in \{1, 2\}, f\_len = 6 \\ i \in \{1, 2, 3\}, f\_len = 8 \\ i \in \{1, 2, 3, 4, 5, 6\}, f\_len = 14 \end{cases} \quad (3)$$

Caso a inequação seja verdadeira para  $f\_len$  igual a 14, então flat2 e flat1 são definidas como “1”. Se não, caso seja verdadeira somente para os outros dois casos, então flat2 é “0” e flat1 é “1”. Em outros casos ambas flat serão “0”. Existe ainda uma análise feita para a aplicação do Filter4, considerando alta variância de borda, a partir de outro parâmetro de limite  $Lim_2$  aplicado na inequação (1) para  $f\_len$  igual a quatro. Esta análise indica o valor da máscara  $hev\_mask$ .

#### 4. RESULTADOS E DISCUSSÃO

O projeto de arquitetura topo em hardware do filtro está representado na Figura 3. A arquitetura é composta por quatro núcleos, sendo que cada núcleo é responsável por filtrar uma linha da borda analisada. Esta escolha de quatro núcleos está relacionada ao menor tamanho de bloco nas partições (4x4). Portanto, a arquitetura recebe um conjunto de 14 amostras por linha (sete de cada lado da borda), e devolve somente as amostras filtradas. Isto quer dizer que, caso o filtro selecionado seja o Filter4, somente as 4 amostras filtradas serão entregues à saída.

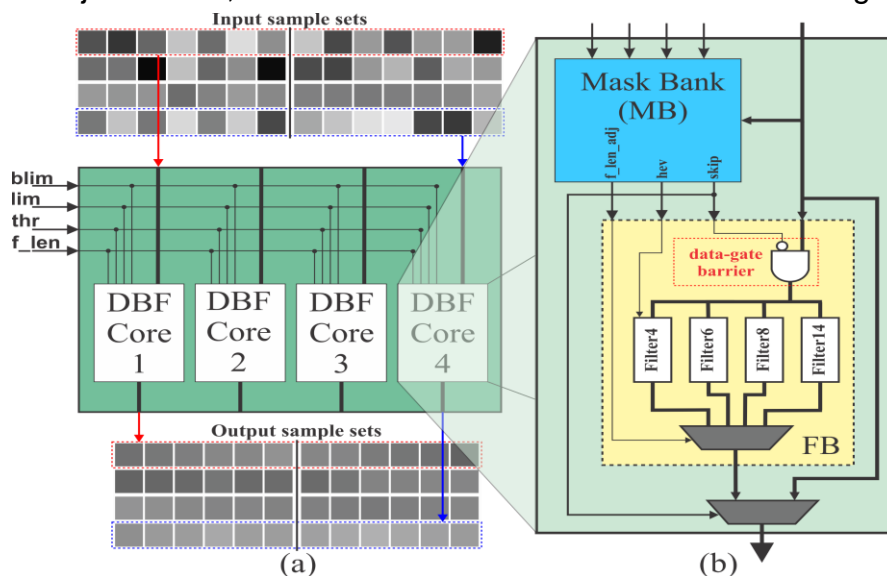


Figura 3 – Arquitetura topo

Outros parâmetros de entrada são as informações necessárias para refinar a escolha de filtro. Tais parâmetros são *blim*, *lim* e *thr*. Todo o cálculo exposto nas equações (1-3) são realizadas no submódulo *Mask Bank* (MB). É no MB também que é gerado o sinal de *skip*, indicando que a arquitetura não filtrará as amostras.

Tendo a escolha do filtro, o submódulo *Filter Bank* (FB) recebe como entrada as amostras selecionadas e as informações provenientes do MB necessárias para aplicação do filtro. Este submódulo possui quatro outros módulos internos: um para cada filtro apresentado anteriormente. A arquitetura opera com todos os filtros e no final seleciona as amostras com base no  $f\_len\_adj$ , ou seleciona as amostras de entrada, caso a máscara *skip* seja verdadeira.

Como primeira etapa de síntese, alguns testes foram realizados para estimar a frequência de operação, dado a complexidade de definição da mesma no contexto de atuação. Para resoluções 4K UHD (4098x2160) em 60 quadros por segundo, foi determinado o pior caso como sendo todas as partições em blocos 4x4, isto é, maior atuação do filtro. Deste pior caso, foi analisado que seria necessária uma frequência de 99,52 MHz para a arquitetura, porém a este pior caso não condiz com o real comportamento dos vídeos. Portanto, foi realizado um experimento sobre sequências de seis vídeos 4K UHD para estimar uma frequência média de operação, contando quantas vezes o filtro atuava. O experimento indicou que seria necessária uma frequência de 16,2 MHz de operação, e os resultados de síntese para as duas frequências estão apresentados na Tabela 1.

Arquitetura	Area (kGates)	Potência Total (mW)	
		Freq. de 99,52 MHz	Freq. de 16,2 MHz
FB	7,15	2,41	0,68
MB	2,13	0,36	0,14
Núcleo DBF	9,84	3,37	0,99
<b>DBF</b>	<b>39,35</b>	<b>13,77</b>	<b>3,96</b>

Tabela 1 – Resultados de Síntese

Conforme indicado na tabela, a arquitetura topo possui uma área de 39,35 kGates para ambas frequências. Porém a arquitetura com a menor frequência de operação possui uma dissipação de potência 71% menor que a do pior caso, sendo uma redução significativa.

## 5. CONCLUSÕES

Este trabalho apresentou uma arquitetura de hardware para o DBF do AV1, com a síntese focando em resoluções 4K UHD, utilizando tecnologia TSMC 40nm. A maior contribuição é a primeira implementação em hardware para uma arquitetura deste filtro, considerando que nenhum outro trabalho focando este filtro foi implementado e publicado até este momento. Como trabalhos futuros, pretende-se implementar todo o laço de filtragem do AV1, e desenvolver uma nova arquitetura para este filtro considerando níveis hierárquicos acima da atual.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

- AOM; AV1 Bitstream & Decoding Process Specification Version 1.0.0 with Errata 1. 2018. AV1. **Alliance for Open Media – Git at Google**. Acessado em: 10 jun. 2019. Disponível em: < <https://aomedia.googlesource.com/aom> >.
- Coronavírus: Facebook, Instagram e Netflix reduzem qualidade de vídeos. **TechTudo**. Acessado em 02 jun. 2020. Disponível em: <<https://www.techtudo.com.br/noticias/2020/03/coronavirus-facebook-instagram-e-netflix-reduzem-qualidade-de-videos.ghml>>.
- CHEN, Y.; MUKHERJEE, D.; Et al. An Overview of Core Coding Tools in the AV1 Video Codec. **2018 Picture Coding Symposium (PCS)**, p.41–45, 2018.
- LEI, Z.; REDDY, S.; CHEREPANOV, V.; DENG, Z.; GPGPU implementation of VP9 in-loop deblocking filter and improvements for AV1 codec. **2017 IEEE International Conference on Image Processing (ICIP)**, Beijing, p.925-929, 2017.
- GRANGE, A.; RIVAZ, P.; HUNT, J.; **VP9 Bitstream & Decoding Process Specification v0.6**. 2016.