

# ARQUITETURA PARA OS FILTROS DE INTERPOLAÇÃO DA FERRAMENTA DE COMPENSAÇÃO DE MOVIMENTO WARPED DO CODIFICADOR AV1

WILLIAM KOLODZIEJSKI<sup>1</sup>; ROBSON DOMANSKI<sup>2</sup>; LUCIANO AGOSTINI<sup>3</sup>

<sup>1</sup>UFPEl – [wkolodziejski@inf.ufpel.edu.br](mailto:wkolodziejski@inf.ufpel.edu.br)

<sup>2</sup>UFPEl – [radomanski@inf.ufpel.edu.br](mailto:radomanski@inf.ufpel.edu.br)

<sup>3</sup>UFPEl – [agostini@inf.ufpel.edu.br](mailto:agostini@inf.ufpel.edu.br)

## 1. INTRODUÇÃO

O AV1 é o primeiro codificador de vídeo da Alliance for Open Media, um consórcio de grandes companhias, como Google, Netflix, Facebook, Cisco, Mozilla e outros. Apesar do AV1 ser capaz de fornecer 20–30% (CHEN, 2018) mais compressão do que seu antecessor, o Google VP9, o esforço computacional do codificador aumentou em mais de 56 vezes (NGUYEN, 2018), o que leva a um aumento no tempo de codificação, decodificação e consumo de energia. Esse aumento de complexidade deve-se às novas e aprimoradas ferramentas, como a *Local Warped Motion Compensation* (LWMC), que utiliza transformações afins para prever movimentos mais complexos que os de translação, como trepidação da câmera, rotação, *zoom* e cisalhamento.

O AV1 é um dos primeiros *codecs* comerciais que exploram transformações afins, já que os sistemas digitais atuais passaram a suportar tal aumento de complexidade. Além do AV1, o padrão *Versatile Video Coding* (VVC), lançado em 2020, também explora transformações afins com uma ferramenta chamada *Affine Motion Compensation* (AMC), porém a implementação dessa é diferente da adotada no AV1. Testes preliminares no conjunto comum de testes do AV1 (NWG, 2020) mostraram ganhos de eficiência de codificação de 1,35–20% ao usar a LWMC (HOLGER, 2020). Então, mesmo com o aumento da complexidade, esses ganhos foram decisivos para definir o uso da LWMC no AV1.

Trabalhos como de PARK (2019) e JIN (2021) exploram métodos para acelerar o processo geral de transformação afim visando o VVC AMC, mas sem implementações de *hardware*. Por outro lado, trabalhos como o de DOMANSKI (2019) e (PENNY, 2019) apresentam projetos de *hardware* para a compensação de movimento (MC) no AV1, mas sem considerar a LWMC. Até onde os autores sabem, nenhuma arquitetura dedicada à LWMC ou AMC pode ser encontrada na literatura. Este trabalho apresenta uma arquitetura dedicada aos filtros de interpolação utilizados na LWMC do AV1. A arquitetura suporta todos os 192 filtros, realizando os cálculos sem o uso de multiplicadores e suportando vídeos em até UHD 8K a 60fps.

## 2. METODOLOGIA

A arquitetura apresentada neste trabalho foi projetada para processar blocos de tamanho 8×8, que é o tamanho suportado pela LWMC. Usando este tamanho de bloco, é possível processar blocos maiores dividindo-os em sub-blocos de 8×8 e depois concatenando-os de volta ao tamanho original. A arquitetura usa um *design* de multifiltro combinacional, chamado *Multifilter Architecture for Local Warped Motion Compensation* (MALW), para interpolar as amostras, explorando o conceito de Multiplicação por Constantes Múltiplas (VORONENKO, 2007), com compartilhamento de sub-expressão entre todos os filtros suportados. Dessa forma, reduz o uso de *hardware*, o consumo de energia e aumenta o *throughput* da arquitetura. A

arquitetura MALW, mostrada na Figura 1, foi projetada para suportar todos os 192 filtros definidos na LWMC usando um único hardware multifiltro, de forma puramente combinatória. A arquitetura permite o processamento paralelo das oito amostras de entrada necessárias para gerar cada amostra filtrada.

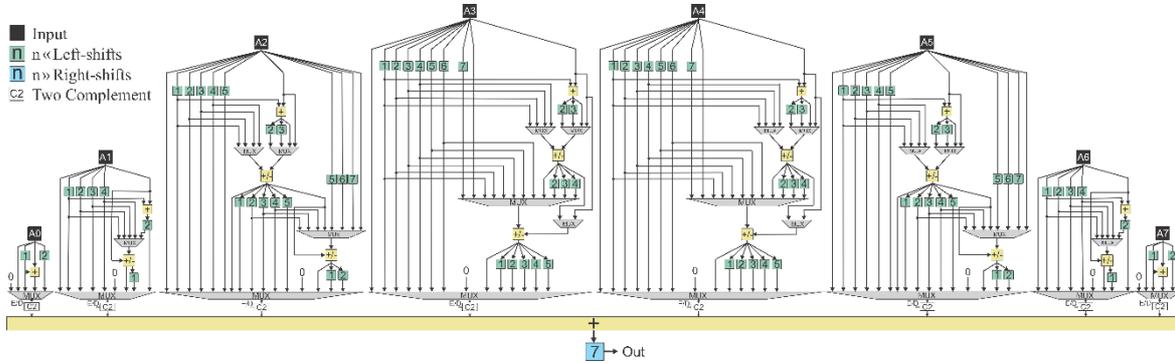


Figura 1- Multifilter Architecture for Local Warped Motion Compensation (MALW).

As entradas A0-A7, mostradas na Figura 1, representam as amostras de entrada dos filtros, e *Out* representa a saída das amostras processadas. Cada amostra pode ser calculada usando deslocadores (caixas verdes) e somadores/subtraidores (caixas amarelas). Os multiplexadores são responsáveis por selecionar qual *tap* do filtro será utilizado e qual operação (soma/desolamento) será realizada, de acordo com o tipo de filtro selecionado.

Uma solução completa de interpolação, chamada *Filter Instance* (FI) e apresentada na Figura 2, foi projetada usando os filtros MALW para processar o bloco 8x8 de entrada. A arquitetura requer uma matriz de entrada de 15x15 amostras (blocos pretos). As instâncias H e V, mostradas na Figura, são idênticas. Os diferentes nomes são usados apenas para destacar se é um filtro vertical ou horizontal.

A solução processa uma linha da matriz de entrada a cada ciclo de *clock*, usando 16 instâncias do MALW. Oito filtros horizontais e oito filtros verticais são conectados através de quatro Cadeias de Registradores Deslocadores (SRC). Assim, cada instância H recebe oito amostras de entrada e gera uma amostra de saída. Cada saída gerada por uma instância H é conectada à SRC, que, por sua vez, é responsável por alimentar corretamente cada instância V. Assim que cada SRC é completamente alimentada, a primeira amostra de saída interpolada é gerada. Assim, essa arquitetura possui uma latência de sete ciclos para iniciar o processamento do filtro V. Desta forma, entre os ciclos 8 e 16, são entregues oito amostras por ciclo de *clock* e, após 16 ciclos, um bloco interpolado 8x8 é gerado.

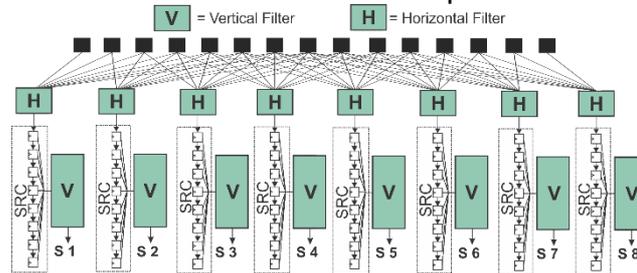


Figura 2- Filter Instance (FI).

### 3. RESULTADOS E DISCUSSÃO

A arquitetura foi descrita em VHDL e sintetizada para células padrão TSMC de 40nm com 1,1V, usando a ferramenta *Cadence RTL Compiler*. Os resultados de energia foram gerados usando a atividade de comutação padrão (20%). A contagem de portas foi calculada com base em NANDs de 2 entradas que representam 0,9408µm<sup>2</sup> para TSMC. A frequência de operação foi definida como 473MHz para

atingir o *throughput* necessário de UHD 8K a 60fps. Os resultados da síntese são apresentados na Tabela 1.

Como não foram encontrados trabalhos na literatura apresentando arquiteturas para a LWMC, não é possível fornecer uma comparação justa de nosso trabalho com trabalhos relacionados. Mesmo assim, apenas para exemplificar as dificuldades com comparações, consideramos três trabalhos, de PENNY (2019), AZGIN (2018) e DOMANSKI (2019) com projetos de *hardware* visando os filtros de interpolação para a ferramenta de MC de diferentes *codecs*.

O trabalho de PENNY (2019) apresenta um interpolador multifiltro, suportando os padrões MPEG-2, MPEG-4, H.264/AVC, HEVC, AVS e AVS2 (um por vez), num total de 20 filtros suportados. Quatro desses filtros têm 8 *taps*, que é o pior caso. A arquitetura processa vídeos em UHD 8K a 60fps. O trabalho de AZGIN (2018) apresenta um interpolador direcionado à MC do VVC, que suporta 15 filtros com 8 *taps*, no pior caso, e processa vídeos em UHD 4K a 95fps. O trabalho de DOMANSKI (2019) suporta todos os 90 filtros definidos na MC do AV1. Nesse caso, os filtros também possuem 8 *taps*, no pior caso. Esse trabalho apresenta duas versões de arquiteturas com diferentes níveis de paralelismo. A solução com o maior paralelismo pode processar vídeos UHD 8K a 30fps.

Entre os trabalhos relacionados, é possível notar que o número de filtros suportados é bastante diferente: 20 em PENNY (2019), 15 em AZGIN (2018), 90 em DOMANSKI (2019) e 192 neste trabalho. Na pior das hipóteses, todos os trabalhos suportam filtros de 8 *taps*, mas o número de filtros é muito diferente. Por exemplo, a solução em PENNY (2019) tem suporte para apenas quatro filtros de 8 *taps*, enquanto nosso *hardware* suporta 192 filtros de 8 *taps*. Essas diferenças afetam os resultados de área e potência. Além disso, os trabalhos relacionados visam diferentes tecnologias e frequências de operação, que também impactam nos resultados de área e energia. Mesmo com essas diferenças e suportando muito mais filtros, pode-se observar que nossa solução consome menos *hardware* que DOMANSKI (2019). Em termos de consumo de energia, se os resultados forem extrapolados para ter a mesma atividade de comutação e a mesma resolução e suporte à taxa de quadros, nossa solução é a que consome menos energia.

Tabela 1- Resultados de Síntese e Comparação com Trabalhos Relacionados

| Arquitetura  | Trabalhos Relacionados |           |           | Este Trabalho |
|--------------|------------------------|-----------|-----------|---------------|
|              | PENNY                  | AZGIN     | DOMANSKI  | FI            |
| Freq. (MHz)  | 471.20                 | 357.00    | 279.90    | 473.00        |
| Gates (K)    | 82.09                  | 11.70     | 141.10    | 119.10        |
| Energia (mW) | 33.79                  | 77.06     | 81.31     | 31.14         |
| Tecnologia   | Nangate 45nm           | TSMC 90nm | TSMC 40nm | TSMC 40nm     |
| Throughput   | 8K 60fps               | 4K 95fps  | 8K 30fps  | 8K 60fps      |
| Filtros      | 20                     | 15        | 90        | 192           |
| Ferramenta   | MC/FME                 | MC        | MC        | LWMC          |
| Padrão       | Vários                 | VVC       | AV1       | AV1           |

#### 4. CONCLUSÕES

Esse trabalho apresentou uma arquitetura para os filtros de interpolação da ferramenta de compensação de movimento *warped* do AV1. A MALW suporta todos os 192 filtros definidos pela LWMC, usando deslocadores e somadores, para reduzir os custos de *hardware*, e explorando uma abordagem de compartilhamento de

sub-expressões, para reduzir a dissipação de energia e aumentar o *throughput*. Esse trabalho também apresentou uma arquitetura completa de interpolação para a LWMC, usando instâncias da arquitetura MALW.

As arquiteturas foram sintetizadas usando a tecnologia de célula padrão TSMC de 40nm e os resultados mostraram que o interpolador LWMC pode processar vídeos em UHD 8K a 60fps com uma dissipação de energia de 31,14mW. Os resultados alcançados são competitivos em comparação com trabalhos relacionados visando filtros de interpolação utilizados em outras ferramentas de codificação de vídeo. Infelizmente, não foram encontradas outras arquiteturas na literatura visando a LWMC do AV1 ou a AMC do VVC, evitando uma comparação mais justa.

## 5. REFERÊNCIAS BIBLIOGRÁFICAS

Grois, D.; Marpe D.; Mulayoff A.; Itzhaky, B.; Hadar, O. Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders. **Picture Coding Symposium (PCS)**, p 394–397, 2013.

Chen, Y. et al. An overview of core coding tools in the AV1 video Codec. **Picture Coding Symposium (PCS)**, p. 41–45, 2018.

Nguyen, T.; Marpe, D. Future video coding technologies: A performance evaluation of AV1, JEM, VP9, and HM. **Picture Coding Symposium (PCS)**, p. 31–35, 2018.

NWG. **Video Codec Testing and Quality Measurement**. Network Working Group, ago, 2020. Acessado em 25 jul. 2022. Online. Disponível em: <https://tools.ietf.org/html/draft-ietf-netvc-testing-09>.

Holger, M.; Ostermann, J. Analysis of Affine Motion-Compensated Prediction in Video Coding. **IEEE Transactions on Image Processing**, v.29, p. 7359–7374, 2020.

Park, S.; Kang, J. Fast Affine Motion Estimation for Versatile Video Coding (VVC) Encoding. **IEEE Access**, v. 7, p. 158075–158084, 2019.

Jin, D. et al. Deep Affine Motion Compensation Network for Inter Prediction in VVC. **IEEE Transactions on Circuits and Systems for Video Technology**, p. 1–1, 2021.

Domanski, R. et al. High-Throughput Multifilter Interpolation Architecture for AV1 Motion Compensation. **IEEE Transactions on Circuits and Systems II: Express Briefs**, v. 66, n. 5, p. 883–887, 2019.

Voronenko, Y.; Püschel, M. Multiplierless multiple constant multiplication. **ACM Transactions on Algorithms**, v. 3, 2007.

Penny, W. et al. High-Throughput and Power-Efficient Hardware Design for a Multiple Video Coding Standard Sample Interpolator. **J. Real-Time Image Process**, v. 16, n. 1, p. 175–192, 2019.

Azgin, H. et al. A Reconfigurable Fractional Interpolation Hardware for VVC Motion Compensation. **2018 21st Euromicro Conference on Digital System Design (DSD)**, p. 99–103, 2018.