



UMA SOLUÇÃO EM HARDWARE PARA A PREDIÇÃO INTRA-QUADRO DO CODIFICADOR DE VÍDEO AOMEDIA VIDEO 1

LUIZ NETO¹; MARCEL CORREA^{1,2}; DANIEL PALOMINO¹;
LUCIANO AGOSTINI¹; GUILHERME CORREA¹

¹*Video Technology Research Group – Universidade Federal de Pelotas*

²*Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense*
{lcmdcneto, mmcorrea, dpalomino, agostini, gcorrea}@inf.ufpel.edu.br

1. INTRODUÇÃO

A codificação de vídeo é uma área amplamente pesquisada, recentemente impulsionada pela crescente demanda por vídeos digitais de ultra-alta definição (UHD). Como o vídeo tem um papel importante em nosso dia a dia, a maior parte do tráfego de consumo da internet já está relacionado a ele e deve crescer para aproximadamente 82% de todo o tráfego de consumo da Internet até 2022 (CISCO, 2017). Durante décadas, esse problema foi abordado por organismos internacionais de padronização por meio do desenvolvimento de padrões de codificação de vídeo. Exemplos recentes são o H.264 *Advanced Video Coding* (AVC) (WIEGAND et al, 2003) e o *High Efficiency Video Coding* (HEVC) (SULLIVAN et al, 2012), lançados em 2003 e 2013, respectivamente. Ambos os padrões foram desenvolvidos por um grupo de especialistas em vídeo do ITU-T e ISO/IEC. No entanto, o licenciamento desses padrões como produtos é um problema. Enquanto o H.264/AVC tem um limite máximo para os custos anuais de licenciamento, o HEVC não. Além disso, o custo total de licenciamento do HEVC pode ser até 16 vezes maior, por unidade, quando comparado ao seu antecessor. Neste contexto, diversas empresas se uniram e criaram a *Alliance for Open Media* (AOM), um consórcio da indústria com mais de 30 empresas do ramo tecnológico, que visa desenvolver formatos de vídeo eficientes e livres de royalties.

O primeiro codificador desenvolvido, lançado em junho de 2018, recebeu o nome de *AOMedia Video 1* (AV1) (CHEN et al, 2018). Desde então, espera-se que o AV1 seja adotado pela maioria das plataformas no futuro próximo. Entretanto, embora o codificador de referência AV1, o *libaom* (RIVAZ; HAUGHTON, 2018), atinja uma eficiência de compressão muito melhor em comparação com as iniciativas antecessoras livres de royalties, o alto custo computacional de várias ferramentas de codificação proíbe seu uso para compressão em tempo real. Portanto, soluções para acelerar a codificação AV1 são de extrema importância. Existem várias maneiras de acelerar a codificação de vídeo e entre o uso de soluções em hardware para ferramentas presentes no codificador é uma das mais eficientes.

Este trabalho propõe uma arquitetura de hardware dedicada para a predição intra-quadro utilizando os modos direcionais do AV1, incluindo os processos de filtragem de borda e upsampling que ocorrem antes a própria predição. A arquitetura desenvolvida visa processar resoluções até ultra-alta definição (UHD) em tempo real.

2. REFERENCIAL TEÓRICO

O processo de codificação do AV1 segue o mesmo fluxo de codificação híbrido de padrões modernos, composto pelas predições inter/intra-quadro, transformação residual e quantização, codificação de entropia e filtragem. Para cada quadro codificado seguindo o formato do AV1, o codificador de referência

libaom divide em superblocos (SBs) de amostras, que podem assumir um tamanho de 64×64 ou 128×128 pixels, conforme configuração do software. Então, cada SB pode ser dividida em nove maneiras diferentes ou manter seu tamanho original. Divisões do SB que resultam em partições quadrada podem ser divididas recursivamente até os menores tamanhos de bloco possíveis, que é 4×4. Estes blocos podem ter os seguintes tamanhos: 64×64, 64×32, 32×64, 64×16, 16×64, 32×32, 32×16, 16×32, 32×8, 8×32, 16×16, 16×8, 8×16, 16×4, 4×16, 8×8, 8×4, 4×8, e 4×4. Na predição intra-quadro, para cada tamanho de bloco, o AV1 suporta vários modos não direcionais e direcionais para permitir a predição com o menor erro possível. A predição direcional permite que o codificador realize a predição utilizando amostras de referência de blocos previamente reconstruídos acima do bloco a ser predito (*AboveRow*) e à esquerda do mesmo (*LeftCol*). Isso é realizado em três etapas: (i) filtragem das amostras de referência, (ii) *upsampling* das amostras de referência e (iii) o próprio processo de predição.

O processo de filtragem dos *arrays* de referência visa suavizar suas amostras, adicionando uma maior precisão ao processo de predição. Quatro forças de filtro, variando de zero a três, podem ser usadas neste processo, onde zero significa não realizar o processo de filtragem. As Equações (1-3), respectivamente, descrevem como o filtro as intensidades de um a três são aplicadas às amostras de referência. Nas equações, *R* é o *array* de referência, *i* é o índice da amostra a ser filtrada e *s* é a amostra filtrada resultante. O processo de *upsampling* serve para dobrar a quantidade de amostras de referência utilizadas no processo de predição, amostras são criadas entre as amostras originais, conforme mostra a Equações (4-5). Onde *U* é o *array* que contém o conteúdo original do *arrays* de referência e a função *Clip* trata os casos em que *aux* é menor que zero ou maior que o máximo permitido pela profundidade de bits.

$$s = ((4 \times R_{i-1} + 8 \times R_i + 4 \times R_{i+1}) + 8) \gg 4 \quad (1)$$

$$s = ((5 \times R_{i-1} + 6 \times R_i + 5 \times R_{i+1}) + 8) \gg 4 \quad (2)$$

$$s = ((2 \times R_{i-2} + 4 \times R_{i-1} + 4 \times R_i + 4 \times R_{i+1} + 2 \times R_{i+2}) + 8) \gg 4 \quad (3)$$

$$aux = ((-U_{i-1} + (9 \times U_i + 9 \times U_{i+1}) - U_{i+2}) + 8) \gg 4 \quad (4)$$

$$s = Clip(aux) \quad (5)$$

AV1 suporta 56 modos direcionais na predição intra, compostos de oito ângulos nominais (45, 67, 90, 113, 135, 157, 180 e 203) e 48 ângulos derivados deles com um deslocamento de -9, -6, -3, +3, +6 e +9 graus. As amostras preditas são obtidas projetando-as no *array* de referência, considerando o ângulo a ser utilizado. Ângulos menores que 90 graus usam o *array AboveRow* como referência para calcular o bloco a ser predito, ângulos maiores que 180 graus usam o *array LeftCol* e ângulos entre 90 e 180 graus usam ambos os *arrays*. A predição de uma única amostra é descrita nas equações (6-8), onde *RefArray* é o *array* que armazena as amostras de referência, *base* é o índice da amostra de referência resultante da projeção da amostra a ser predita e *shift* é um coeficiente usado para realizar a multiplicação pelas amostras do *array* de referência. As equações usadas para determinar a *base* e o *shift* podem ser encontradas em (RIVAZ; HAUGHTON, 2018). Por fim, *Pred_{x,y}* é o valor da amostra predita localizada na posição determinada por *x* e *y*.

$$mult_1 = RefArray_{base} \times (32 - shift) \quad (6)$$

$$mult_2 = RefArray_{base+1} \times shift \quad (7)$$

$$Pred_{x,y} = (mult_1 + mult_2 + 16) \gg 5 \quad (8)$$

3. ARQUITETURA DESENVOLVIDA

A arquitetura proposta funciona com SBs de tamanho 64×64 , executa o processo de filtragem das amostras de referência e *upsampling* e a predição dos oito ângulos nominais em paralelo. A saída é uma linha ou coluna predita por ciclo por ângulo nominal. Esta arquitetura é controlada por uma máquina de estados finita, responsável por fazer todas as combinações de partições de bloco permitido pelo AV1, a partir do 64×64 SB, funcionando como uma busca em profundidade até o menor tamanho possível, 4×4 . A Figura 1 mostra o maior nível hierárquico da arquitetura.

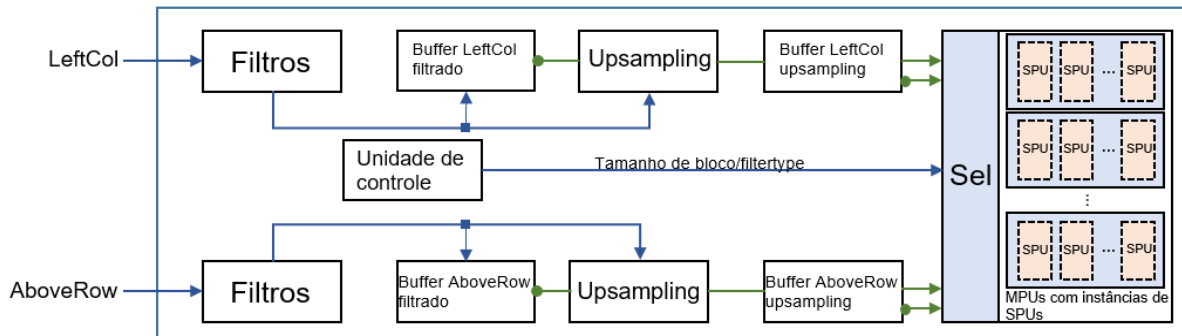


Figura 1. Diagrama de blocos da arquitetura proposta.

Os filtros requerem até três amostras de referência de cada lado da amostra que será filtrada. Portanto, os *arrays* de amostras de referência devem ser estendidos copiando a primeira e a última amostra de referência três vezes. Então, a arquitetura do filtro é capaz de realizar o processo de filtragem dos *arrays* de referência, usando os três tipos de filtros em paralelo, com três buffers para amostras filtradas e um fio para amostras não filtradas como a saída. Um seletor de filtro é usado para selecionar o *LeftCol* filtrado e *AboveRow* filtrado adequadamente para cada *Mode Prediction Unit* (MPU), de acordo com o processo de seleção da força do filtro descrito em (RIVAZ; HAUGHTON, 2018). Para aumentar a quantidade dos *arrays* de referência, a arquitetura de *upsampling* usa equações (4-5) para criar amostras interpoladas localizadas entre as amostras originais. Para realizar a predição de uma única amostra, as equações (6-8) foram implementadas na arquitetura *Sample Prediction Unit* (SPU). Os valores dos coeficientes (*shift*) variam de 0 a 32 e são complementares, ou seja, quando um coeficiente é igual a 15 o outro necessariamente será 17. Por causa disso, 16 diferentes SPUs foram implementadas e instanciadas de acordo com a necessidade de cada MPU. SPUs são usados em uma arquitetura maior, a MPU, que é responsável por organizar as amostras de referência que serão calculadas pelas SPUs e para prever uma linha ou coluna por ciclo. Um total de 10 MPUs são implementadas na arquitetura, 6 para os modos normais e 4 para os modos que usam o *upsampling*. O número de SPUs em uma MPU varia de acordo com o ângulo do modo. Além disso, o ângulo do modo também define se uma linha ou coluna do bloco será predita.

4. RESULTADOS E DISCUSSÃO

A arquitetura proposta descrita na seção anterior foi sintetizada para a tecnologia TSMC 40nm usando a ferramenta Cadence Encounter RTL Compiler 14.02. A Tabela 1 mostra os resultados obtidos em termos de dissipação de potência (mW), consumo de energia por amostra predita (nJ / amostra) e a área

do circuito, considerando cada taxa de transferência alvo (fps) e a frequência necessária (MHz) para operação em tempo real.

Os resultados da Tabela 1 mostram que, como esperado, a dissipação de potência aumenta à medida que a frequência aumenta. É interessante notar que embora a potência aumente à medida que a frequência aumenta, a arquitetura apresenta uma boa eficiência energética para o caso 1080@60fps, que requer um consumo de energia de 2,047 nJ/amostra. Isso vem com um pequeno aumento na área do circuito em comparação com o caso 1080p@30fps. A arquitetura proposta é a primeira na literatura capaz de realizar filtragem e *upsampling* dos *arrays* de referência e a predição intra-quadro usando o ângulos nominais do AV1.

Tabela 1. Resultados de síntese.

Taxa de transferência (fps)	Frequência alvo (MHz)	Potência total (mW)	Consumo de energia (nj/amostra)	Área (Kgates)
1080p@30fps	215	180,221	1,931	688,897
1080p@60fps	476	382,076	2,047	691,742
4K@30fps	951	823,156	2,205	808,313
4K@60fps	1902	161,383	2,161	821,835

5. CONCLUSÕES

Este trabalho apresentou uma solução em hardware para a predição intra-quadro do codificador AV1, capaz de realizar as operações de pré-processamento de filtragem e *upsampling* dos *arrays* de referência, bem como a predição direcional com todos os ângulos nominais suportados no AV1.

A arquitetura proposta suporta o pior cenário possível de particionar cada SB para o número máximo de blocos, todas as possibilidades de filtro e o uso do *upsampling*. A arquitetura realiza predição intra-quadro em tempo real para vídeos 1080p@60fps a 476 MHz com um custo de energia de 2,047 nJ/amostra. Além disso, vídeos de 4K@30fps e 4K@60fps podem ser processados a 951 MHz e 1902 MHz, com um custo de 2,205 e 2,161 nJ/amostra, respectivamente.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- CISCO. **Cisco Visual Networking Index: Forecast and Methodology**, 2016–2021. Disponível em: <<https://www.cisco.com/c/en/us/solutions/collateral/serviceprovider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf>>.
- WIEGAND, T.; SULLIVAN, G. J.; BJØNTEGAARD, G.; LUTHRA, A. Overview of the H.264/AVC video coding standard. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.13, n.7, p.560–576, 2003.
- SULLIVAN, G. J.; OHM, J.; HAN, W.; WIEGAND, T. Overview of the High Efficiency Video Coding (HEVC) Standard. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.22, n.12, p.1649–1668, 2012.
- CHEN, Y. et al. An Overview of Core Coding Tools in the AV1 Video Codec. In: PICTURE CODING SYMPOSIUM (PCS), 2018., 2018. **Anais. . .**, 2018. p.41–45.
- RIVAZ, P.; HAUGHTON, J. **AV1 Bitstream Decoding Process Specification**. Disponível em: <<https://aomedia.org/av1/specification/>>.