

ARQUITETURAS DE HARDWARE PARA OS FILTROS DE INTERPOLAÇÃO DA ESTIMAÇÃO DE MOVIMENTO AFFINE DO PADRÃO VVC

DENIS MAASS¹; MURILO PERLEBERG¹; VLADIMIR AFONSO²;
LUCIANO AGOSTINI¹; MARCELO PORTO¹

¹Universidade Federal de Pelotas – Video Technology Research Group (ViTech)

²Instituto Federal Sul-rio-grandense Campus Pelotas

denismaass7@gmail.com, {mrperleberg, vafonso, agostini, porto} @inf.ufpel.edu.br

1. INTRODUÇÃO

Com a popularização da internet e a disseminação de dispositivos móveis, a demanda por vídeos digitais tem aumentado ano após ano. Contudo, devido a grande quantidade de dados que é inerente a este tipo de conteúdo, são necessárias técnicas de compressão destes dados para que possa ser possível armazenar e transmitir vídeos de forma eficiente. Atualmente, o estado da arte em codificação de vídeo é o padrão *Versatile Video Coding* (VVC), lançado no ano de 2020 (GONÇALVES, 2021).

Para se obter uma codificação de vídeo eficiente, é fundamental explorar as redundâncias temporais, isto é, a semelhança entre dois quadros de um vídeo. Neste contexto, a Estimação de Movimento (EM) é a principal técnica utilizada. A EM emprega algoritmos de busca para encontrar similaridades entre os blocos do quadro que está sendo codificado e os blocos dos quadros de referência. O bloco com a melhor correspondência, ou bloco predito, é referenciado a partir de um vetor de movimento (VM) que caracteriza o deslocamento translacional do bloco predito em relação ao bloco atual (GONÇALVES, 2021).

Embora a EM já tenha sido explorada em quase todos os padrões de codificação de vídeo anteriores, no padrão VVC a inovação é a adoção da Estimação de Movimento *Affine* (EMA), que permite representar não apenas os movimentos translacionais, como também movimentos de rotação e *zoom* (LI et al., 2018).

A EMA é aplicada em blocos com um tamanho de pelo menos 16x16 amostras e pode ser de 4 ou 6 parâmetros, sendo que o de 4 parâmetros utiliza os VMs de dois pontos de controles do bloco, como mostra a Figura 1 (a). Já o de 6 parâmetros utiliza os VMs de três pontos de controle, como pode ser visto na Figura 1 (b) (GONÇALVES, 2021).

Para reconstruir um bloco codificado com a EMA, é necessário dividi-lo em blocos de tamanho 4x4 amostras para que cada sub-bloco possa ser reconstruído individualmente. Para isso, o VM da amostra central do sub-bloco é calculado a partir dos 4 ou 6 parâmetros, e este VM será utilizado por todas as amostras do sub-bloco (GONÇALVES, 2021). A divisão do bloco em sub-blocos e os VMs das amostras centrais são mostrados na Figura 2 (c).

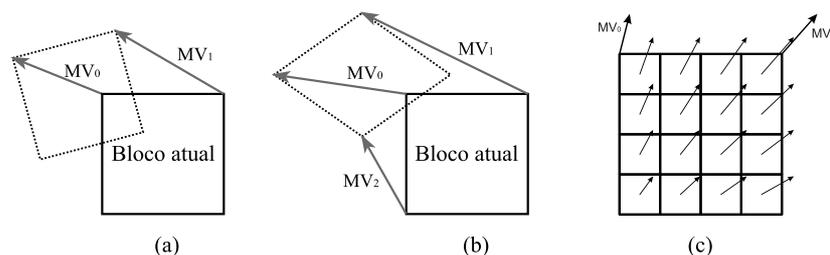


Figura 1. Representação do *Affine*. (a) 4 Parâmetros; (b) 6 Parâmetros; e (c) Divisão do bloco em vários sub-blocos para a Reconstrução *Affine*

Uma vez que o movimento ocorrido entre dois frames não está limitado apenas a posições inteiras, os vetores dos sub-blocos podem possuir precisão fracionária e, portanto, uma importante etapa dentro da EMA é a interpolação fracionária, que consiste na geração de amostras intermediárias em posições fracionárias entre as amostras inteiras (AFONSO et al., 2016).

Para gerar as amostras fracionárias e, conseqüentemente, reconstruir os blocos codificados com a EMA, o VVC utiliza filtros do tipo *6-tap* (BROSS et al., 2021), o que significa que para a geração de uma amostra interpolada são necessárias 6 amostras de entradas, as quais serão ponderadas por seus respectivos coeficientes. A eq. (1) mostra uma equação genérica para um filtro de interpolação F_N , onde c_x representam os coeficientes do filtro e A_x as amostras de entrada, que são as amostras ao redor da amostra a ser gerada.

$$F_N = (c_0A_{-2} + c_1A_{-1} + c_2A_0 + c_3A_1 + c_4A_2 + c_5A_3 + 32) / 64 \quad (1)$$

No total, o VVC utiliza 15 diferentes filtros (BROSS et al., 2021), sendo o valor dos coeficientes diferentes para cada filtro e variando de acordo com a distância da amostra de entrada para a amostra fracionária a ser gerada. Com isso, alcança-se uma resolução de *pixel* de 1/16, o que significa que entre dois *pixels* inteiros vizinhos horizontalmente podem ser geradas 15 novas amostras fracionárias, o mesmo ocorre entre dois *pixels* verticalmente vizinhos. Além disso, ainda podem ser geradas mais $15 \times 15 = 225$ amostras diagonais fracionárias.

Contudo, essa interpolação de amostras é de uma elevada complexidade, fazendo com que a codificação tome muito tempo (GONÇALVES, 2021), impossibilitando que soluções de *software* sejam utilizadas em aplicações que demandem codificação em tempo real. Para lidar com esse problema uma abordagem possível é a codificação através de um *hardware* dedicado que, diferentemente da codificação por *software*, conta com um processador específico para realizar a codificação, deixando a CPU livre para executar outras tarefas (KUFA; KRATOCHVIL., 2017).

Neste trabalho serão propostas duas abordagens para a implementação em *hardware* dos 15 filtros utilizados pela EMA do padrão VVC. A primeira implementação utiliza multiplicadores em sua arquitetura, enquanto na segunda é aplicada uma estratégia de otimização onde os multiplicadores são substituídos por circuitos de somas e deslocamentos binários.

2. METODOLOGIA

Em uma visão de alto nível, ambos modelos de arquiteturas recebem 6 amostras de 10 *bits* na entrada, essas amostras são ponderadas de acordo com a equação do filtro, e então geram uma amostra interpolada também de 10 *bits*.

Para exemplificar a metodologia utilizada e as diferenças entre as duas abordagens será utilizado como exemplo o filtro F_8 , cuja equação é dada pela eq. (2).

$$F_8 = (3A_{-2} - 11A_{-1} + 40A_0 + 40A_1 - 11A_2 + 3A_3 + 32) / 64 \quad (2)$$

A primeira abordagem utiliza multiplicadores e é uma aplicação direta da equação do filtro, onde as 6 amostras de entrada são multiplicadas por seus respectivos coeficientes e somadas juntamente com uma constante igual a 32 e, por fim, é feita a divisão por 64. A Figura 2 (a) apresenta a arquitetura desenvolvida por essa abordagem.

Já na segunda abordagem, todos os fatores de multiplicação da equação são decompostos em somas de potências de 2, assim todas as multiplicações podem ser executadas por somas e deslocamentos binários, que requerem menos *hardware* para serem implementadas. Para isso, a eq. (2) pode ser reescrita como na eq. (3).

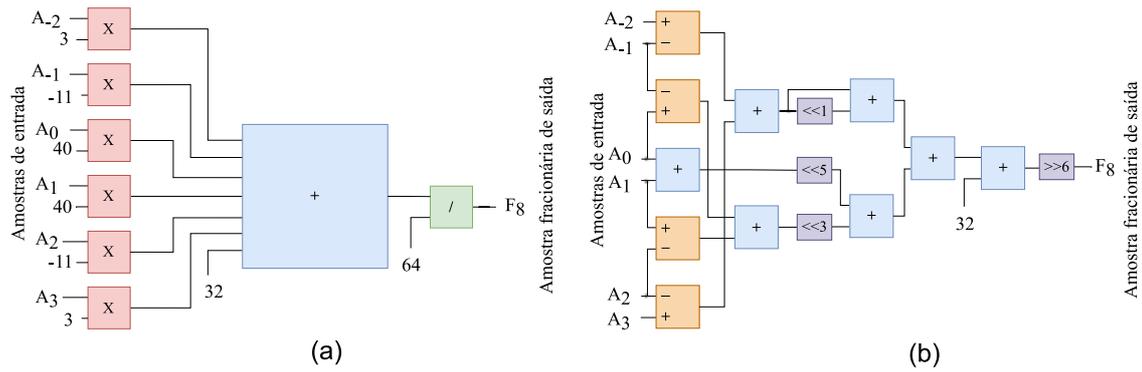


Figura 2. Arquitetura do filtro F_8 . (a) multiplicadores; e (b) somas e deslocamentos

$$F_8 = [(2+1)A_{-2} - (8+2+1)A_{-1} + (32+8)A_0 + (32+8)A_1 - (8+2+1)A_2 + (2+1)A_3 + 32] / 64 \quad (3)$$

Reorganizando a eq. (3), de modo que todos os termos que são multiplicados pelos mesmos coeficientes fiquem agrupados, e sabendo que uma multiplicação por dois pode ser implementada pelo deslocamento binário em um *bit* para a esquerda, obtemos a eq. (4). A eq. (4) representa a equação de será de fato adotada pela segunda abordagem, e sua implementação pode ser vista na Figura 2 (b).

$$F_8 = [(A_{-2} - A_{-1} + A_3 - A_2) + (A_{-2} - A_{-1} + A_3 - A_2) \ll 1 + (A_0 - A_{-1} + A_1 - A_2) \ll 3 + (A_0 + A_1) \ll 5 + 32] \gg 6 \quad (4)$$

3. RESULTADOS E DISCUSSÃO

As arquiteturas dos 15 filtros foram desenvolvidas utilizando as duas abordagens apresentadas. As 30 arquiteturas foram então descritas em linguagem VHDL e validadas utilizando a ferramenta *ModelSim* da Intel. As arquiteturas foram então sintetizadas em ASIC com a ferramenta *Cadence RTL Compiler* e utilizando a biblioteca de células padrão de 40nm da TSMC.

O número de portas lógicas utilizadas e a potência dissipada por cada filtro desenvolvido, considerando a frequência de 100MHz, é mostrado na Tabela 1.

Tabela 1. Resultados de síntese

Filtro	Multiplicadores		Somas e deslocamentos		Ganho (%)	
	Potência (mW)	Nº de portas	Potência (mW)	Nº de portas	Potência (mW)	Nº de portas
1	0,222	916	0,196	876	-11,71%	-4,35%
2	0,238	984	0,204	893	-14,29%	-9,30%
3	0,226	912	0,207	942	-8,41%	3,22%
4	0,310	1206	0,255	1053	-17,74%	-12,71%
5	0,287	1109	0,283	1114	-1,39%	0,48%
6	0,303	1211	0,249	1028	-17,82%	-15,11%
7	0,334	1300	0,277	1086	-17,07%	-16,48%
8	0,320	1247	0,251	940	-21,56%	-24,60%
9	0,341	1318	0,275	1086	-19,35%	-17,59%
10	0,305	1216	0,249	1028	-18,36%	-15,48%
11	0,286	1134	0,271	1089	-5,24%	-4,01%
12	0,294	1188	0,257	1076	-12,59%	-9,42%
13	0,225	911	0,209	933	-7,11%	2,42%
14	0,244	1003	0,204	893	-16,39%	-11,02%
15	0,213	913	0,197	876	-7,51%	-4,03%
Média	0,277	1105	0,239	994	-13,60%	-10,00%

Na Tabela 1, pode-se observar que os filtros desenvolvidos pela abordagem de somas e deslocamentos conseguem atingir uma redução de 21,56% e 24,60% em potência dissipada e número de portas lógicas, respectivamente. Na média dos 15 filtros, a abordagem com somas e deslocamentos apresenta uma redução de potência em 13,60%, e requerem 10% menos portas lógicas. Essa redução significativa se deve ao fato de multiplicadores demandarem somadores e portas lógicas adicionais para a sua implementação.

Também foi realizada uma investigação preliminar a respeito da frequência máxima em que podem operar as duas implementações propostas. Os resultados preliminares foram obtidos sintetizando o filtro F_8 até sua máxima frequência. Foi possível observar que a abordagem com multiplicadores é capaz de alcançar uma frequência de 951 MHz, enquanto a arquitetura com somas e deslocamentos pode operar em até 1169 MHz, o que significa um aumento de 22,9% na frequência máxima ao empregar a abordagem de somas e deslocamentos.

4. CONCLUSÕES

Foram apresentadas duas implementações de *hardware* para os 15 filtros de interpolação da EMA do padrão VVC, sendo uma delas uma aplicação direta da equação do filtro, utilizando multiplicadores, e outra otimizada, utilizando somas e deslocamentos, de modo a se obter uma melhor eficiência energética e aproveitamento de área. Vale ressaltar que esta é a primeira implementação destes filtros na literatura e, em seguida, eles serão integrados na reconstrução da EMA.

5. REFERÊNCIAS BIBLIOGRÁFICAS

LI, L. et al. An Efficient Four-Parameter Affine Motion Model for Video Coding. **IEEE Transactions on Circuits and Systems for Video Technology**, v.28, n.8, p.1934-1948, 2018.

AFONSO, V. et al. Hardware Implementation for the HEVC Fractional Motion Estimation Targeting Real-Time and Low-Energy. **Journal of Integrated Circuits and Systems**, v.11, n.2, p.106-120, 2016.

KUFA, J; KRATOCHVIL, T. Software and hardware HEVC encoding. **International Conference on Systems, Signals and Image Processing (IWSSIP)**, p.1-5, 2017.

BROSS, B. et al. Overview of the Versatile Video Coding (VVC) Standard and its Applications. **IEEE Transactions on Circuits and Systems for Video Technology**, v.31, n.10, p.3736-3764, 2021.

GONÇALVES, P.H.R. **Um Esquema Rápido Baseado em Aprendizado de Máquina para a Predição Interquadros do Codificador de Vídeo VVC**. 2021. Dissertação (Mestrado em Ciência da Computação) - Programa de Pós-graduação em Computação do Centro de Desenvolvimento Tecnológico da Universidade Federal de Pelotas.