

## REUSO DE DADOS E OPERAÇÕES VISANDO O PROCESSAMENTO DE MÚLTIPLOS TAMANHOS DE BLOCO EM UM CIRCUITO DEDICADO PARA IME DO HEVC

MURILO PERLEBERG<sup>1</sup>; VLADIMIR AFONSO<sup>2</sup>;  
LUCIANO AGOSTINI<sup>1</sup>; BRUNO ZATT<sup>1</sup>; MARCELO PORTO<sup>1</sup>

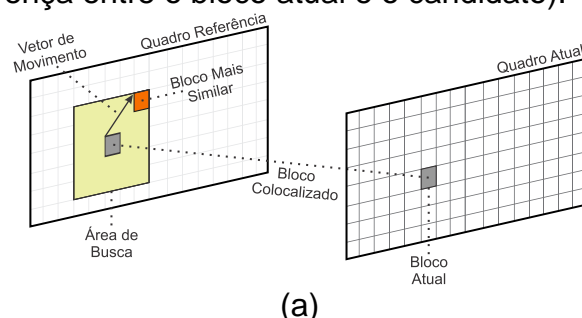
<sup>1</sup>Universidade Federal de Pelotas – {mrperleberg, agostini, zatt, porto}@inf.ufpel.edu.br

<sup>2</sup>Instituto Federal Sul Rio-Grandense – vafonso@inf.ufpel.edu.br

### 1. INTRODUÇÃO

Vídeos digitais estão cada vez mais presentes no cotidiano das pessoas, seja em forma de entretenimento (em serviços de *streaming* como *Youtube* e *Netflix*), como também em reuniões por vídeo-chamada devido a atual pandemia. Buscando melhorar a eficiência do armazenamento e transmissão desses vídeos, diversas técnicas de compressão são aplicadas sobre os vídeos digitais. Assim, o padrão *High Efficiency Video Coding* (HEVC) (SULLIVAN, 2012) é o padrão estado-da-arte atualmente, englobando diversas técnicas de compressão que permitem obter uma alta eficiência de compressão.

Durante a codificação de um vídeo digital, o vídeo é dividido em vários quadros (imagem em um único instante de tempo). Então, cada quadro é dividido em Unidades de Codificação em Arvore (CTUs) de tamanho 64x64. Cada CTU pode ser dividida em diversos blocos menores nos quais são empregadas as técnicas de compressão. Dentre as técnicas empregadas no HEVC, a Estimação de Movimento Inteira (IME) é a ferramenta mais complexa, porém indispensável em qualquer padrão de codificação devido a sua grande eficiência. A Figura 1-(a) representa o funcionamento da IME. A IME busca representar o bloco do quadro que está sendo codificado utilizando blocos de quadros referência (quadros já codificados). Para isso, a IME precisa localizar qual bloco candidato em uma área de busca reduzida (pertencente ao quadro anterior) é mais similar ao bloco sendo codificado. O resultado da IME é o vetor da posição do melhor bloco candidato, e o resíduo (diferença entre o bloco atual e o candidato).



$$SAD = \sum_{(x,y) \in B} |C_{(x,y)} - R_{(x,y)}|$$

Figura 1. (a) Processamento da IME; (b) Formula do SAD.

Para avaliar cada candidato, a métrica SAD representada na Figura 1-(b) é aplicada sobre cada candidato, calculando sua diferença para o bloco atual, amostra por amostra, e então somando o absoluto de todas as diferenças obtidas. São diversos os fatores que contribuem para a grande complexidade da IME, como o grande número de candidatos existentes dentro da área de busca, além de que no HEVC cada CTU pode ser dividida em até 24 diferentes tamanhos de bloco menores. Além disso, os algoritmos empregados por padrão dão total liberdade para que diferentes blocos candidatos possam ser avaliados para cada possível

bloco, dificultando uma avaliação eficiente através do compartilhamento de dados e de operações.

Devido a sua alta complexidade, comumente a IME é implementada em circuitos dedicados, de forma a obter a compressão de vídeos em alta resolução e em tempo real. Porém, muitos dos trabalhos propondo soluções arquiteturais para a IME focam apenas na unidade de processamento, negligenciando os acessos a memória necessários para o funcionamento da arquitetura, os quais são frequentemente mais relevantes do que os resultados da unidade de processamento. Sendo assim, este trabalho apresenta um circuito dedicado para a IME, com foco em obter acessos mais eficazes a memória através do compartilhamento de dados e de operações realizadas.

## 2. METODOLOGIA

O trabalho pode ser dividido em duas etapas. Na primeira delas o algoritmo a ser implementado pela arquitetura foi definido e avaliado quanto a sua eficiência na compressão de vídeos. Isso é, foram definidas estratégias para obter um acesso mais regular a memória, e também aplicadas restrições no número total de candidatos a serem avaliados de forma a garantir o processamento de vídeos em alta resolução em tempo real. Além disso, nessa primeira etapa o processamento de todos os tamanhos de bloco foram sincronizados, de forma que o maior tamanho de bloco do HEVC (64x64) é o que define quais os candidatos que serão avaliados nos blocos menores. As estratégias propostas foram todas avaliadas considerando as Condições Comuns de Teste do HEVC (SHARMAN, 2018) para medir o seu impacto na eficiência da compressão de vídeos, onde foi obtido um aumento médio na métrica BD-rate (BJONTEGAARD, 2018) de 1,94%, o que representa uma pequena perda na eficiência da codificação. Detalhes específicos de todas as estratégias empregadas no algoritmo foram ocultadas por conta do espaço disponível.

Já a segunda etapa do trabalho é a implementação da arquitetura utilizando todas as estratégias do algoritmo avaliado. A arquitetura completa está representada na Figura 2-(a), sendo que ela possui três unidades principais: O controle que implementa o algoritmo para escolher quais candidatos serão requisitados da memória para serem avaliados, além de duas unidades de processamento.

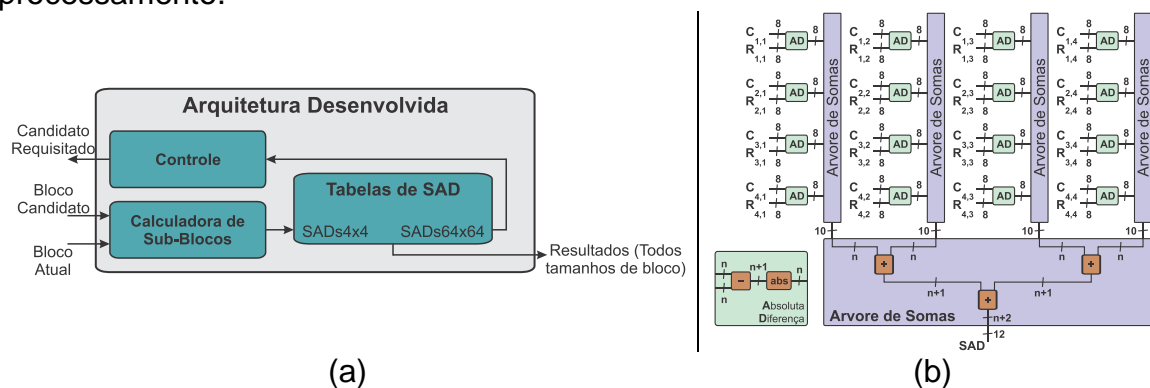


Figura 2. Arquitetura Desenvolvida: (a) Arquitetura Topo; (b) Unidade Calculadora de Sub-Blocos.

A primeira unidade de processamento é a Calculadora de Sub-Blocos. Essa unidade recebe as amostras do bloco atual e do candidato definido pelo controle e divide essas amostras em sub-blocos 4x4. Para cada sub-bloco é então calculado

o valor do SAD utilizando a arquitetura visível na Figura 2-(b), a qual possui 16 unidades de Absoluta Diferença e 5 Árvores de Soma para calcular o valor do SAD de cada sub-bloco. Posteriormente, o SAD dos sub-blocos é repassado para a Tabela de SAD, a segunda unidade de processamento da arquitetura. Essa unidade é responsável por agrupar o SAD de sub-blocos vizinhos, compondo e resultando assim no SAD de todos os 24 tamanhos de bloco diferentes considerando o candidato que está sendo avaliado. Após isso, o SAD do bloco 64x64 é repassado para o controle da arquitetura, que o utiliza para definir o próximo candidato a ser avaliado.

A arquitetura foi desenvolvida com 16 linhas de paralelismo, isso é, ela é capaz de processar 16 linhas do bloco candidato a cada ciclo de clock. Sendo assim, a arquitetura precisa de 4 ciclos de clock para processar cada candidato. Ao considerar o pior caso avaliado (SHARMAN, 2018) em que o algoritmo proposto escolhe até 444 candidatos para serem avaliados, além de alguns ciclos onde a arquitetura está ociosa aguardando decisões do controle, a arquitetura precisa de 1850 ciclos para processar cada CTU do vídeo. Assim, para vídeos 2160p@30fps, que possuem 61,2k CTUs, a arquitetura precisa operar a 113,2 MHz.

### 3. RESULTADOS E DISCUSSÃO

A arquitetura foi descrita em VHDL e sintetizada para tecnologia ASIC utilizando a ferramenta *Cadence RTL Compiler* e a biblioteca de células de 40nm da TSMC. Os resultados de síntese mostraram que para o processamento de vídeos 2150p@30fps, a arquitetura dissipa 81,74mW. Além disso, os resultados mostraram que a arquitetura desenvolvida é capaz de atingir a frequência de até 566,1MHz, sendo assim capaz de processar vídeos de até 2160p@150fps. Outros resultados e características são apresentadas na Tabela I, juntamente com os resultados de alguns trabalhos relacionados.

Tabela I – Resultados comparativos da arquitetura proposta

|                              | Fan (2018)  | Gu (2019)    | Kim (2020)   | Este Trabalho    |
|------------------------------|-------------|--------------|--------------|------------------|
| Algoritmo                    | Diamond     | Diamond      | TZS          | TZS Sincronizado |
| Área de Busca                | 192x192     | 96x96        | 192x192      | 192x192          |
| Tamanhos de Bloco            | Todos       | Quadrados    | Todos        | Todos            |
| BD-rate (%)                  | -0,5        | 0,55         | 0,17         | 1,94             |
| Tecnologia ASIC              | 65nm        | TSMC 65nm    | 65nm         | TSMC 40nm        |
| Área (Portas NAND2)          | 489,4k      | 225,7k       | 439,8k       | 269k             |
| Taxa máxima de processamento | 2160p@30fps | 2160p@139fps | 2160p@120fps | 2160p@150fps     |
| Potência (mW) (2160p@30fps)  | 128,5       | N.A.         | 145,66       | 81,74            |

Os trabalhos da literatura propõem diferentes técnicas para reduzir a complexidade da IME e assim obter arquiteturas mais eficientes. FAN (2018) propõe uma arquitetura em diamante que avalia muito mais blocos que o algoritmo TZS padrão, porém com menor quantidade de decisões do que o algoritmo TZS. Devido a essa grande quantidade de candidatos, FAN (2018) consegue obter um pequeno ganho na eficiência de compressão. Além disso, a quantidade de recursos do seu circuito e a potência dissipada são maiores do que a do nosso circuito.

A arquitetura de GU (2019) também adota um algoritmo em diamante que avalia diversos candidatos, porém ele adota uma área de busca reduzida e suporta

apenas blocos de tamanho quadráticos para reduzir a complexidade da IME. A arquitetura de GU (2019) requer uma menor área do que a do trabalho proposto, porém GU (2019) não apresentou uma análise do consumo de energia da arquitetura proposta.

Já KIM (2020) propõe uma arquitetura com um certo nível de reuso de operações entre blocos vizinhos, enquanto limita o número de ciclos a ser dedicado para o processamento de cada CTU. Embora ele atinja uma eficiência de compressão melhor do que a do trabalho proposto, a arquitetura de KIM (2020) requer uma maior área e possui um maior consumo energético do que o trabalho proposto. Por último, pode ser notado que a arquitetura proposta atinge a maior taxa de processamento dentre os trabalhos relacionados.

A arquitetura proposta possibilita uma grande redução nos acessos a memória. Considerando que todos os 24 tamanhos de bloco do HEVC processem os mesmos candidatos, a redução final é de até 23/24 acessos visto que cada candidato só será solicitado uma única vez. Contudo, ao considerar que o algoritmo padrão é independente para cada tamanho de bloco e, portanto, diferentes candidatos podem ser solicitados para cada bloco a ser avaliado, essa redução na comunicação com a memória pode ser ainda maior. Essa grande redução nos acessos a memória é muito vantajosa para a redução do consumo energético do *chip* de memória, visto que cada acesso a memória possui um alto consumo. O trabalho completo explora com mais detalhes o consumo energético da unidade de processamento e da memória empregada. Esse trabalho completo está no processo final de escrita e visa a publicação na revista *IEEE Transactions on Circuits and Systems for Video Technology*.

#### 4. CONCLUSÕES

Esse trabalho propõe uma arquitetura para a ferramenta IME do HEVC, capaz de processar todos os tamanhos de bloco através do sincronismo de todos os blocos de cada CTU. Esse sincronismo possibilita o reuso de dados e de operações durante o processamento, além da redução nos acessos realizados a memória.

#### 5. REFERÊNCIAS BIBLIOGRÁFICAS

- SULLIVAN, G. J. et al. Overview of the High Efficiency Video Coding (HEVC) Standard. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 22, n. 12, p. 1649-1668, 2012.
- SHARMAN, K. et al. Common Test Conditions. **JCTVC-AF1100**, Ljubljana, 2018.
- BJONTEGAARD, G., Improvements of the BD-PSNR model, **VCEG-AI11**, 2018.
- FAN, Y. et al. A Hardware-Oriented IME Algorithm for HEVC and its Hardware Implementation, **IEEE Transactions on Circuits and Systems for Video Technology**, v. 28, n. 8, p. 2048-2057, 2018.
- GU, C. et al. A Micro-Code-Based Hardware Architecture of Integer Motion Estimation for HEVC, **IEEE International Conference on Very Large Scale Integration**, Peru, 2019.
- KIM, T. et al. Fast Hardware-Based IME With an Idle Cycle and Computational Redundancy Reduction, **IEEE Transactions on Circuits and Systems for Video Technology**, v. 30, n. 6, p. 1732-1744, 2020.