Arquitetura de SAD Energeticamente Eficiente Utilizando Blocos Comprimidos na Estimação de Movimento

Vítor Costa; Murilo Perleberg; Luciano Agostini; Marcelo Porto

Universidade Federal de Pelotas – Video Technology Research Group (ViTech)
[vscosta, mrperleberg, agostini, porto]@inf.ufpel.edu.br

1. INTRODUÇÃO

Atualmente, o tráfego de conteúdo de vídeo na internet alcançou um novo propósito. No passado, os meios de vídeo eram utilizados principalmente para entretenimento, conectar amigos distantes ou registrar momentos únicos. Nos dias de hoje, os conteúdos em vídeo tornaram-se altamente diversificados em seus propósitos: empresas de streaming são preferidas em relação à televisão tradicional pelo público (EVENS et. al, 2023) e o marketing digital junto com as redes sociais exploram cada vez mais a criação de vídeos curtos que captam rapidamente a atenção do espectador (DASH, 2024). Todas essas tendências evidenciam a necessidade de uma maneira eficiente de lidar com conteúdo de vídeo.

No entanto, o processo de codificação de vídeo é uma tarefa altamente intensiva em recursos computacionais, exigindo diversas limitações e otimizações nos padrões de codificação para possibilitar sua implementação em uma arquitetura de hardware dedicada e eficiente, que esteja alinhada com restrições físicas, como dissipação de energia e calor, área de circuito, frequência de operação e acessos à memória, entre outros. Entre todas as etapas presentes nos codificadores de vídeo atuais a Estimação de Movimento (ME) se destaca devido à sua enorme complexidade, por conta do elevado número de cálculos e avaliações que o codificador precisa executar. A etapa de ME consome até 60,7% do tempo total de codificação (GRELLERT et. al, 2016) e, contém, no mínimo, 29,5% dos acessos à memória no codificador HEVC (MATIVI et. al, 2016).

Este artigo propõe o uso de blocos comprimidos na ME, por meio de duas arquiteturas, reduzindo assim o tamanho/potência da memória interna da ME e também a dissipação de energia pela unidade operativa da ME.

2. MOTIVAÇÃO

A ME tem como objetivo estimar o deslocamento de uma Unidade de Predição (PU), um bloco de amostras de um quadro que está sendo codificado, em relação ao Bloco Candidato (CB) mais similar de um quadro de referência já codificado. Embora ofereça capacidades expressivas de compressão, a ME é, de longe, a etapa mais intensiva em recursos nos codificadores de vídeo atuais, pois deve calcular o critério de similaridade SAD para todos os CBs dentro de uma Área de Busca (SA) delimitada para encontrar o CB mais adequado para a PU atual.

A compressão de dados tanto da PU quanto dos CBs pode ser realizada como uma estratégia para reduzir os requisitos de memória em um hardware dedicado para a ME. Em (FAN et. al, 2018), um algoritmo de compressão de blocos é proposto para comprimir PUs e CBs de 4x4, reduzindo os 128 bits necessários (4 x 4 x 8 bits) para apenas 32 bits. Ao comprimir os blocos, é possível não apenas reduzir o tamanho da memória necessária, mas também diminuir significativamente

a potência relacionada à memória interna de ME em pelo menos 75%, devido a um número reduzido de acessos.

Além dos benefícios relacionados à memória, outra abordagem também pode ser explorada com o uso de blocos comprimidos durante o processo de ME. O algoritmo proposto em (FAN et. al, 2018), que será detalhado na Seção III, traz o benefício de permitir a reutilização de dados durante os cálculos de SAD. Se menos valores forem utilizados para representar um bloco 4x4 inteiro, alguns cálculos de SAD se tornam redundantes. O algoritmo define 16 modos de compressão e, de acordo com a combinação desses modos de compressão selecionados, algumas operações podem ser ignoradas durante o processo de SAD. Ao monitorar os modos de compressão de cada bloco, é possível desativar dinamicamente algumas unidades operativas na árvore de SAD, reduzindo ainda mais o consumo de energia da unidade de processamento da ME.

3. ARQUITETURAS PROPOSTAS

Este artigo propõe duas arquiteturas de hardware distintas: o Block Compressor Unit (BCU) e o Configurable SAD Tree Calculator (CSTC).

O BCU é responsável por comprimir blocos 4x4 (PUs e CBs) com amostras de 8 bits (totalizando 128 bits) em um vetor de apenas 32 bits. O algoritmo de compressão empregado pelo BCU foi proposto em (FAN et. al, 2018), onde os autores definem 16 modos de compressão a serem avaliados sobre o bloco de entrada 4x4. Os modos podem ser caracterizados como verticais, horizontais ou de quadrante, de acordo com quais amostras são selecionadas para representar o bloco de entrada.

A arquitetura proposta do BCU pode ser vista na Fig. 1 e é dividida em quatro estágios. O primeiro estágio é responsável por gerar os blocos reconstruídos 4x4, reutilizando quatro valores amostrais do bloco de entrada para cada um dos 16 modos. Em seguida, o segundo estágio calcula os valores de SAD entre o bloco comprimido reconstruído e o bloco original, para cada um dos 16 modos, utilizando 16 árvores de SAD. O terceiro estágio realiza a seleção do melhor modo de compressão, com base no menor SAD encontrado no estágio anterior, a partir de uma árvore multiplexadora. Finalmente, o quarto estágio constrói o vetor de 32 bits com base no melhor modo analisado, concatenando um prefixo de 4 bits, referente ao melhor modo de compressão, com os valores truncados das quatro amostras selecionadas.

Conforme mencionado anteriormente, o uso de dados comprimidos nas PUs nos CBs pode reduzir o número de computações necessários durante o cálculo do SAD. Para explorar os benefícios do bloco comprimido, este artigo propõe o *Configurable SAD Tree Calculator* (CSTC). A implementação do CSTC é mostrada na Fig. 2 e na Fig. 3. O CSTC consiste de uma árvore de SAD típica de bloco 4x4 com 31 operadores (somas e subtrações), mas com uma unidade de controle chamada de *Mode Analyzer* (MA). O MA avalia quais operadores devem ser utilizados para obter o SAD, com base nos quatro bits mais significativos (modo de compressão) de cada um dos dois blocos de entrada comprimidos. O MA da arquitetura busca explorar a reutilização de dados proveniente destas correlações, implementando a lógica seletora apresentada na Fig. 3, gerando uma palavra de controle. Cada bit da palavra de controle é direcionado a cada uma das 16 unidades *multiplexed_diff* da arquitetura vista na Fig. 2. Cada unidade de *multiplexed_diff* é

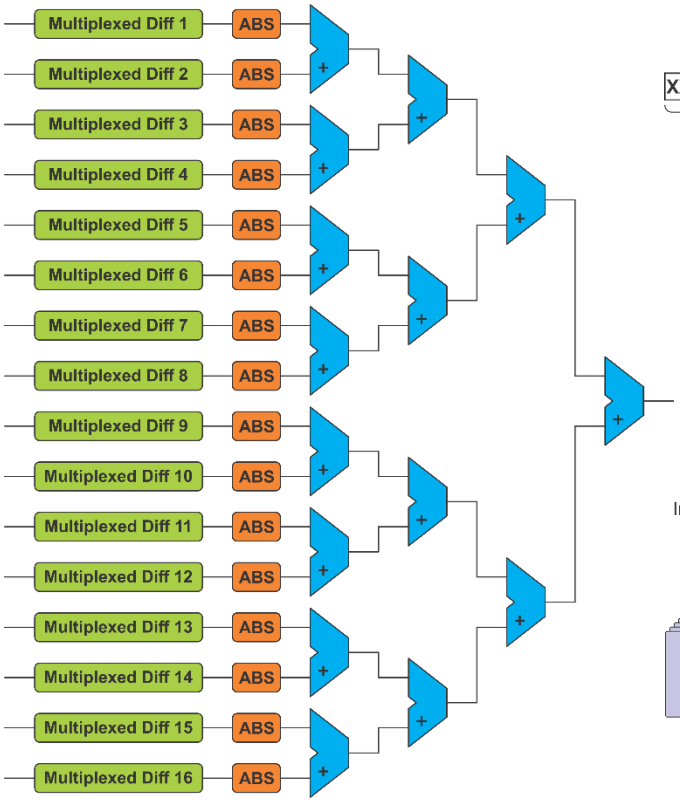

Fig. 2. Arquitetura do CSTC

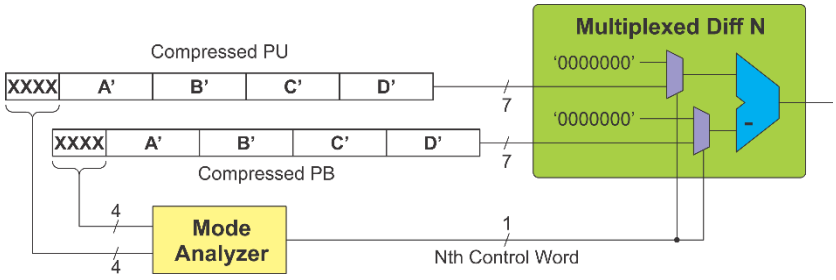

Fig. 1. Arquitetura do BCU

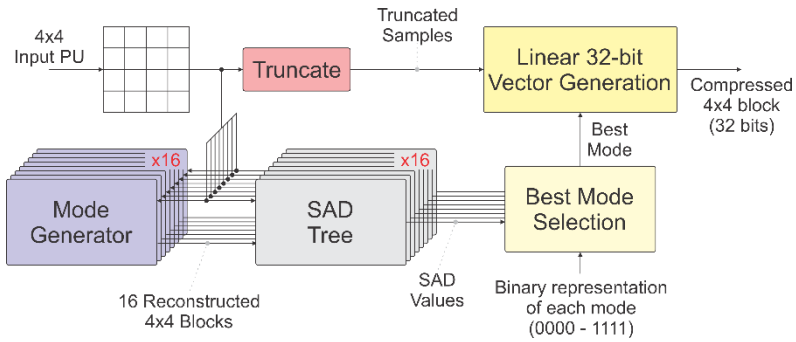

Fig. 3. Visão detalhada do MA e do *multiplexed_diff*

composta por dois MUXes 2:1 seguidos um subtrator. Quando o bit de controle é "0", o *multiplexed_diff* força o valor "0" nas entradas do subtrator, consequentemente, desativando a referida unidade. Essa estratégia permite uma redução na atividade de chaveamento da arquitetura, diminuindo assim sua dissipação de potência.

Frente as combinações de modos possíveis o CSTC pode operar em três cenários: o melhor caso, com apenas 4 *multiplexed_diff* ativos, o caso intermediário com 8 *multiplexed_diff* e o pior caso com 16 *multiplexed_diff*.

4. RESULTADOS DE SÍNTESE

As arquiteturas propostas foram descritas em VHDL e sintetizadas utilizando a ferramenta RTL Compiler em uma biblioteca de células de tecnologia 40nm da TSMC. A frequência operacional de 370MHz foi utilizada para ambas as arquiteturas, correspondendo à frequência máxima suportada pela arquitetura do BCU, a qual possui o maior caminho crítico entre as duas arquiteturas. Para validar as arquiteturas e gerar os resultados de potência, foi utilizado um arquivo (VCD) contendo dez mil entradas geradas com chaveamento aleatório.

O *overhead* de potência e de área imposto pelo BCU foram 85.02mW e 52.97 Kgates respectivamente, enquanto os resultados da síntese da arquitetura CSTC são apresentados na Tabela 1, onde a dissipação de potência foi avaliada para os diferentes caso em que o CSTC pode operar. A Tabela 1 também apresenta os resultados da síntese para uma árvore de SAD padrão, que tem o mesmo comportamento do CSTC no pior cenário.

Considerando a árvore de SAD padrão como referência para os resultados, é possível observar que o CSTC requer mais área (cerca de 14,7%) devido ao hardware de controle adicional. Em relação aos resultados de potência, o pior

cenário do CSTC apresenta uma dissipação de potência maior (cerca de 6,8%), conforme esperado. No entanto, no cenário intermediário e no melhor cenário apresentam reduções na dissipação de potência de 35,2% e 58,6%, respectivamente. Para encontrar as reduções de potência do CSTC em um cenário realista, é necessário descobrir as ocorrências dos modos de compressão selecionados de cada PU e CB durante o processo de codificação de vídeo.

Tabela 1 – Resultados de Síntese do CSTC para cada caso de operação

Arquitetura	Área (Kgates)	Potência dinâmica (mW)	Potência total (mw)	Variação da potência total (%)
SAD Tree padrão	18.89	1.26	1.26	-
CSTC (melhor caso)	21.67	0.52	0.52	-58.6%
CSTC (caso intemediário)	21.67	0.81	0.82	-35.2%
CSTC (pior caso)	21.67	1.34	1.35	+6.8%

5. CONCLUSÕES

Este artigo propôs uma abordagem eficiente em termos de potência para o cálculo de SAD sobre blocos comprimidos na ME, a qual pode ser aplicada a qualquer padrão contemporâneo de codificação de vídeo. As arquiteturas propostas podem produzir uma redução de 75% tanto no tamanho da memória interna da ME quanto em sua dissipação de potência, além de uma redução de potência de até 58,6% na unidade de processamento de SAD.

6. REFERÊNCIAS BIBLIOGRÁFICAS

T. Evens, A. Henderickx and P. Conradie. **Technological affordances of video streaming platforms: Why people prefer video streaming platforms over television,** 2023 European Journal of Communication, 39, pp. 1-19, doi: 10.1177/02673231231155731.

Dash. **Video marketing statistics for your 2024 campaings**, Retrieved May 30, 2024: https://www.dash.app/blog/video-marketing-statistics

M. Grellert, S. Bampi and B. Zatt. **Complexity-scalable HEVC encoding**, *2016 Picture Coding Symposium (PCS)*, Nuremberg, Germany, 2016, pp. 1-5, doi: 10.1109/PCS.2016.7906356.

A. Mativi, E. Monteiro and S. Bampi. **Memory access profiling for HEVC encoders**, 2016 IEEE 7th Latin American Symposium on Circuits & Systems (LASCAS), Florianopolis, Brazil, 2016, pp. 243-246, doi: 10.1109/LASCAS.2016.7451055.

Y. Fan, L. Huang, B. Hao and X. Zeng, "A Hardware-Oriented IME Algorithm for HEVC and Its Hardware Implementation," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 8, pp. 2048-2057, Aug. 2018, doi: 10.1109/TCSVT.2017.2702194.