

## ARQUITETURA PARA A ESTIMAÇÃO DE MOVIMENTO EM MÚLTIPLOS TAMANHOS DE BLOCOS A PARTIR DO CÁLCULO DE SADs PARCIAIS

**MATEUS MELO; GUSTAVO SMANIOTTO; LUCIANO AGOSTINI; BRUNO ZATT;  
MARCELO PORTO**

*Universidade Federal de Pelotas – Grupo de Arquitetura e Circuitos Integrados  
{msdmelo, ghsmaniotto, agostini, zatt, porto}@inf.ufpel.edu.br*

### 1. INTRODUÇÃO

No presente mercado de produtos eletrônicos, há um predomínio de dispositivos com suporte à captura e manipulação de vídeos digitais. Entretanto estes dispositivos apresentam limitações, tanto em eficiência energética como em armazenamento de dados, que tornam inviável o armazenamento e/ou transmissão de vídeos digitais sem o uso de mecanismos de compressão, dada a grande volume de dados a serem tratados. Para facilitar a manipulação de vídeos digitais, é utilizado um codificador de vídeo, que tem como objetivo representar a mesma informação (vídeo) de forma comprimida.

O padrão de codificação de vídeo estado considerado como arte é o HEVC (*High Efficient Video Coding*) (UTI-T, 2013), sendo o padrão capaz de atingir as maiores taxas de compressão, quando comparado aos demais padrões existentes no mercado. Entretanto, para obter elevadas taxas de compressão são necessárias etapas de codificação, cada vez mais complexas. Dentre estas etapas, a Estimação de Movimento (ME) é aquela que proporciona aos maiores ganhos em compressão, ao custo de representar o maior percentual no tempo de codificação. A ME é a etapa encarregada de identificar redundâncias temporais, ou seja, redundâncias entre quadros temporalmente vizinhos, trabalhando em nível de bloco em um quadro a ser codificado, realizando a busca por um bloco mais similar em um quadro previamente codificado (quadro de referência).

Existem diversos algoritmos de ME, com distintos padrões de busca adotados. Estes algoritmos podem ser classificados como: ótimo ou rápidos. O algoritmo Full Search (FS) é o algoritmo ótimo, pois compara todos os blocos presentes dentro de uma área de busca (região menor que a resolução do quadro atual, presente em um quando já codificado). Os algoritmos rápidos dispõem de heurísticas de aceleração para a escolha do bloco mais similar, que possibilitam uma redução no número de blocos a serem comparados, sem um padrão regular de busca.

De outro modo, o algoritmo FS não apresenta dependências de dados entre os blocos a serem comparados, além de ser regular em relação ao padrão de comparação. Sendo assim, este algoritmo proporciona uma implementação em hardware simplificada quando comparada a algoritmos considerados rápidos.

A SAD (Soma das Diferenças Absolutas) é o critério de similaridade mais adotado na literatura para identificar o bloco mais similar na área de busca. Sua popularidade está no fato de serem utilizadas operações aritméticas simples (absoluto, adição, soma) para seu cálculo, além de não existir dependências de dados entre os blocos a serem comparados. Estes fatos, facilitam a implementação do critério de SAD de forma paralela em hardware.

Este trabalho apresenta uma arquitetura paralela de hardware para o cálculo de SAD de blocos de tamanho 8x8 utilizando o algoritmo FS. Os valores de SAD de blocos de tamanhos 8x8 podem ser somados para a formação de valores de SAD de blocos de tamanhos maiores utilizando apenas uma soma entre dois valores de SAD de blocos menores para se calcular o valor de SAD de um bloco de tamanho superior, reduzirá o número de total operações realizadas na etapa de ME.

## 2. METODOLOGIA

A estratégia adotada neste trabalho é descrita por ZHENYU (2007) para o padrão antecessor ao padrão HEVC, com um menor número de tamanhos de blocos à serem utilizados. O tamanho de bloco 8x8 pixels é adotado como bloco base pois este tamanho de bloco está presente em todos os padrões de codificação de vídeo atuais, além de ser um tamanho adequado ao desenvolvimento da arquitetura de múltiplos tamanhos de blocos a partir da soma (agrupamento) de valores de SAD. A implementação deste trabalho visa a partir de blocos de tamanho 8x8 pixels, compor blocos até o tamanho de bloco 64x64 pixels (maior tamanho de bloco suportado no padrão HEVC).

Considerando o padrão HEVC, a arquitetura implementada não suporta os tamanhos de blocos assimétricos e blocos de tamanho 4x8 e 8x4, dado que estes tamanhos de blocos não são frequentemente utilizados no padrão (AFONSO, 2015), além de que, conforme AFONSO (2015), a remoção desses tamanhos de bloco do codificador apresentam baixas perdas em eficiência de codificação.

O algoritmo FS é adotado neste trabalho pois identifica o bloco mais similar presente na área de busca, além da regularidade no padrão de busca do algoritmo facilitar a estratégia do agrupamento de valores de SAD. Foi adotada uma área de busca de 38x38 pixels para o tamanho de bloco 8x8, ao redor do bloco co-localizado no quadro de referência, resultando assim em 961 blocos a serem comparados. Considerando a estratégia de agrupamento de valores de SAD, este número de blocos a serem comparados será o mesmo para todos os tamanhos de blocos a serem processados.

O fluxo de cálculos de SAD adotado neste trabalho é apresentado na Figura 1.A, na qual *Cálculo SAD8x8* (caixa cinza) é o módulo mais complexo da arquitetura implementada, pois calcula os valores de SAD de um bloco utilizando árvores de SAD. Como pode ser visto na Figura 1.A os valores de SAD dos blocos de tamanho  $N \times N$  (largura x altura) são utilizados para calcular os valores de SAD de blocos candidatos de tamanho  $2N \times N$  e  $N \times 2N$ . Ambos os tamanhos de bloco são compatíveis para obter-se o valor de SAD de um bloco de tamanho  $2N \times 2N$ , sendo que neste trabalho foi adotado o tamanho de bloco  $N \times 2N$  para este fim.

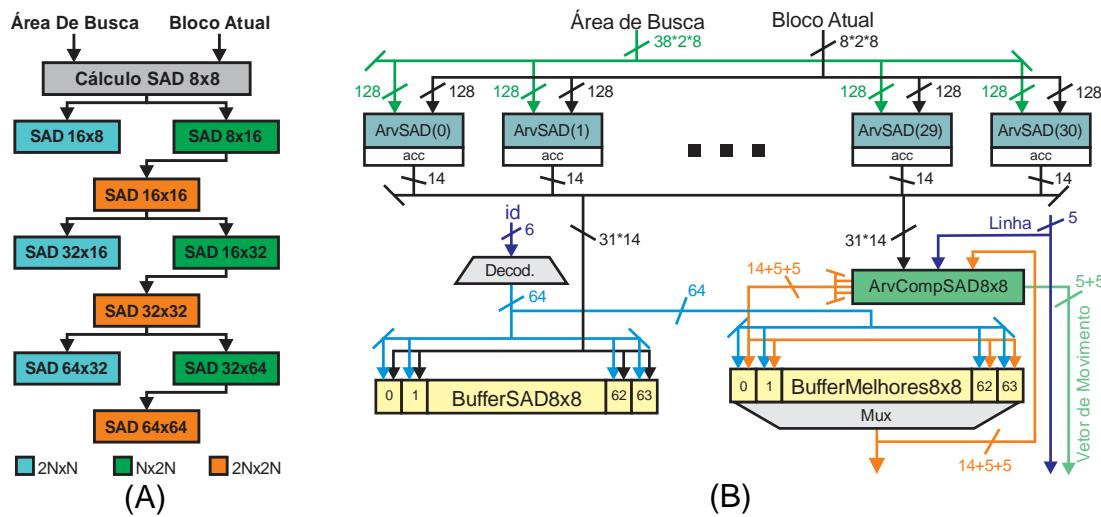


Figura 1. Fluxograma da solução (A) e módulo de cálculo de SAD de blocos 8x8 (B)

O diagrama de blocos do módulo de cálculo de SAD para blocos de tamanho 8x8 é apresentado na Figura 1.B, conforme pode ser visto, esta arquitetura tem como entrada: duas linhas de amostras da área de busca ( $2 \times 38$ ) e duas linhas de amostras do bloco à ser codificado ( $2 \times 8$ ); e como saídas o vetor de movimento indicando a posição do bloco mais similar. Esta arquitetura é composta de seis

módulos: *ArvSAD* (árvore de SAD, composta de 16 operadores de módulo, 16 subtratores e 15 somadores), *Comp.ArV8x8* (árvore de comparadores, 31 valores em paralelo), *BufferSAD8x8* (armazenamento dos valores de SAD para serem utilizados na etapa de agrupamento), *BufferMelhores8x8* (armazena o melhor valor de SAD para cada bloco 8x8), *mux* e *decod*.

Para o processamento dos dados na arquitetura é necessário dividir um quadro do vídeo em blocos de tamanho 64x64, e cada um destes blocos em blocos de tamanho 8x8. Após, cada um desses blocos será dividido em duas linhas para ter-se como entrada na arquitetura; além de que, a área de busca também será processada a cada duas linhas. Sendo assim são necessários quatro ciclos de *clock* para obter-se o valor de SAD de uma linha de (31) blocos à serem comparados.

Assim, a arquitetura processará a primeira linha de blocos a serem comparados presentes na área de busca com o primeiro bloco de tamanho 8x8, e após, com os demais blocos de tamanho 8x8. Em seguida, a mesma ideia será utilizada, entretanto, utilizando a segunda linha de blocos a serem comparados presentes na área de busca para todos os blocos de tamanho 8x8. O processo será feito para todos os blocos de tamanho 8x8 e em toda área de busca.

Os módulos utilizados para agrupamento são muito similares ao módulo apresentado na Figura 1.B, sua diferença principal se dá na substituição de *ArvSAD* por somadores de n-bits de acordo com o tamanho de bloco; de modo a serem utilizadas nove arquiteturas (caixas azuis, verdes e laranjas na Figura 1.A), uma para cada tamanho de bloco. Outra diferença, se refere a eliminação do *bufferSAD* para as arquiteturas referentes aos tamanhos de bloco  $2N \times N$  (blocos azuis na Figura 1.A), pois os valores de SAD para estes tamanhos de blocos não são utilizados para calcular o valor de SAD de blocos maiores. Além disso, o número de *bufferSAD* e *bufferMelhores* é inversamente proporcional ao tamanho do bloco, pois, quanto maior o tamanho do bloco, menos blocos deste tamanho estão presentes em um bloco de tamanho 64x64, por exemplo, 64 blocos de tamanho 8x8, 4 blocos de tamanhos 32x32.

Os módulos de agrupamento possuem dependência de dados de acordo com o atual bloco 8x8 que está sendo processado. A Figura 2 apresenta estas dependências temporais de processamento para o agrupamento de valores de SAD, na qual os quadrados em cinza representam que o agrupamento pode ser feito para tal tamanho de bloco, de acordo com o bloco de tamanho 8x8 a ser processado.

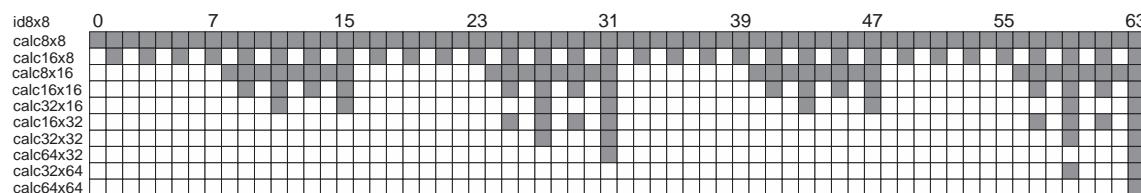


Figura 2. Dependências temporais para a etapa de agrupamento de valores de SAD

A arquitetura completa processa cada linha de SAD em quarto ciclos para um bloco de tamanho 8x8. Deste modo, são necessários 64x8 ciclos de *clock* para processar a primeira linha de SAD dos 64 blocos de tamanho 8x8; entretanto é necessário processar as 31 linhas de SAD presentes na área de busca. Assim, são necessários 7684 ciclos para definir o bloco mais similar do primeiro bloco de tamanho 8x8. Então, 7936 ciclos de *clock* são necessários para o processamento completo de todos os blocos de tamanho 8x8, já com o bloco mais similar para cada tamanho de bloco.

A arquitetura apresentada neste trabalho, foi descrita na linguagem VHDL, validada através de simulações com dados reais utilizando a ferramenta Mentor Graphics ModelSim. Os resultados de síntese e performance foram extraídos para o dispositivo Stratix V 5SGXMABN3F45I4 utilizando a ferramenta Quartus Prime 16.0, e resultados de estimativa potência dissipada foram obtidos a partir do netlist (obtido pela ferramenta Quartus Prime 16.0) referente à arquitetura, utilizando a ferramenta Altera PowerPlay, integrada à ferramenta Quartus Prime 16.0.

### 3. RESULTADOS E DISCUSSÃO

Conforme os resultados apresentados na Tabela 1, esta arquitetura apresenta uma frequência máxima de operação de 199,68 MHz, de modo que é possível processar 49 quadros por segundo (QPS) na resolução Full HD (1920x1080 pixels), sendo superior à taxa de QPS necessária para sensação de movimento nesta resolução (30 QPS).

Tabela 1. Resultados de síntese e potência – Stratix V 5SGXMABN3F45I4

Síntese		Potência – Full HD (4 quadros)	
<b>ALMs</b>	69.418 (19%)	-	-
<b>Registradores</b>	63.132	<i>E/S</i> (mW)	32,52
<b>Pinos</b>	838 (79%)	<i>Dinâmica</i> (mW)	55,90
<b>Frequência</b>	199,68 MHz	<i>Estática</i> (mW)	1.192,00
<b>QPS – Full HD</b>	49	<b>Total</b> (mW)	1.280,41

Os resultados de potência foram extraídos a partir de entradas reais de quatro quadros de um vídeo de resolução Full HD, dissipando 1.280,42mW de potência. Entretanto, vale ressaltar que por se tratar de um dispositivo FPGA existem outros componentes de hardware funcionando em paralelo com a arquitetura, dissipando principalmente potência estática. Deste modo, os resultados mais influenciados pelo funcionamento da arquitetura referem-se à dissipação de potência dinâmica e de E/S, representando apenas 7% da potência total dissipada pelo dispositivo.

### 4. CONCLUSÕES

Este artigo apresentou uma arquitetura paralela para Estimação de Movimento para codificadores de vídeo, etapa com maior percentual no tempo de codificação, descrita em VHDL. Esta arquitetura possui suporte à maioria dos tamanhos de blocos adotados pelos principais padrões de codificadores de vídeo presentes no mercado atual, incluindo o padrão considerado como estado da arte, o padrão HEVC. Esta abordagem de reuso de operadores é encontrada na literatura apenas para padrões de codificação de vídeo antecessores ao padrão HEVC.

Os resultados de performance e potência, para a arquitetura, foram obtidos utilizando dados reais, extraídos de um de um vídeo Full HD, demonstrando que a arquitetura desenvolvida é capaz de processar vídeos de alta resolução em tempo real.

### 5. REFERÊNCIAS BIBLIOGRÁFICAS

ITU-T, International Telecommunication Union. **Recommendation ITU-T H.265**: High efficiency video coding. Abril, 2013.

Liu, Zhenyu, et al. "Hardware-efficient propagate partial sad architecture for variable block size motion estimation in H. 264/AVC." In **ACM Great Lakes symposium on VLSI**. ACM, 2007.

V. Afonso et al. "Memory-Aware and High-Throughput Hardware Design for the HEVC Fractional Motion Estimation". In **SBCCI**, Salvador, 2015, p. 11.