

ARQUITETURA DE HARDWARE EFICIENTE PARA A RESOLUÇÃO DE SISTEMAS DE EQUAÇÕES LINEARES E CÁLCULO DE ΔMV NA PREDIÇÃO AFFINE DO VVC

DENIS MAASS¹; MARCELLO MUÑOZ¹; MURILO PERLEBERG¹;
LUCIANO AGOSTINI¹; MARCELO PORTO¹

¹Universidade Federal de Pelotas – Video Technology Research Group (ViTech)
{dlmaass, mmunoz, mrperleberg, agostini, porto}@inf.ufpel.edu.br

1. INTRODUÇÃO

Nos últimos anos, impulsionado pela popularização da internet e dos serviços de *streaming*, observamos um substancial aumento na demanda por vídeos digitais (MUÑOZ et al., 2023). Além disso, à medida que as resoluções de vídeo crescem, a necessidade de lidar com volumes cada vez maiores de dados para representar esse conteúdo torna-se evidente. Assim, para armazenar e transmitir vídeos digitais de maneira eficaz, torna-se necessário recorrer a técnicas de compressão de dados. Neste cenário, o padrão Versatile Video Coding (VVC) (BROSS et al., 2021), lançado em 2020, destaca-se como o estado da arte em compressão de vídeo.

O padrão VVC aplica inúmeras técnicas que visam diminuir a quantidade de dados necessários para representar um vídeo. Uma das mais importantes consiste em explorar as redundâncias temporais, isto é, a semelhança entre os quadros do vídeo. Esta técnica é chamada de Estimção de Movimento (EM), e consiste em empregar algoritmos de busca para encontrar similaridades entre os blocos do quadro que está sendo codificado e os blocos dos quadros de referência. Onde 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 (MUÑOZ et al., 2023).

A EM já foi empregada em diversos padrões de codificação anteriores ao VVC, no entanto, essa técnica se limita apenas à representação de movimentos translacionais. Neste contexto, a novidade no padrão VVC fica por conta da implementação da Estimção de Movimento *Affine* (EMA), que permite representar movimentos mais complexos, como rotação e *zoom* (LI et al., 2018). A EMA pode ser realizada nos modos de 4 ou 6 parâmetros, onde na configuração de 4 parâmetros são utilizados dois Vetores de Movimentos (VMs), conforme ilustrado na Figura 1 (a). Já na configuração de 6 parâmetros, a EMA faz uso dos VMs de três pontos de controle, como exemplificado na Figura 1 (b).

Para encontrar o conjunto de VMs que melhor descreve o movimento na AME, o VVC emprega um algoritmo iterativo chamado *Affine Inter-Mode*, que refina um conjunto inicial de VMs, baseado nos valores de erro e de gradiente do bloco gerado a cada iteração (MARTINA, 2022). Neste processo, para gerar o próximo

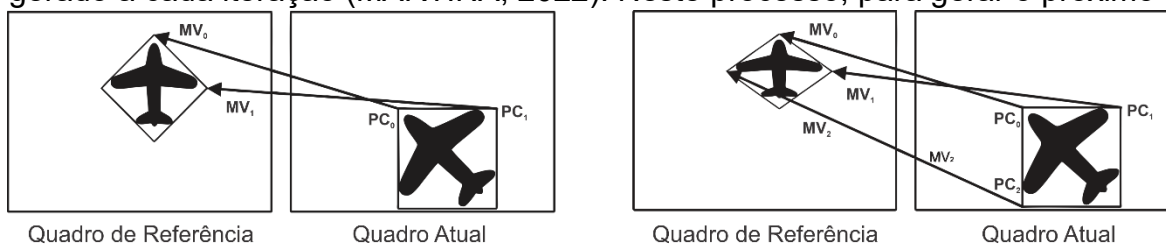


Figura 1: Modos *Affine*: (a) 4 parâmetros e (b) 6 parâmetros

conjunto de MVs a ser testado, é necessária a resolução de um sistema de equações lineares, que dará origem a um conjunto de ΔMVs para a próxima iteração. A resolução deste sistema é feita a partir da aplicação do método de Eliminação Gaussiana com Pivotamento Parcial (RODRIGUES DA SILVA, 2018), que consiste em transformar a matriz associada ao sistema em uma forma triangular superior.

A implementação da Eliminação Gaussiana com Pivotamento Parcial em hardware apresenta dificuldades devido ao seu caráter sequencial, à necessidade de múltiplas operações de linha e à escolha de pivôs a cada etapa, o que aumenta a complexidade de controle e a latência do processamento. Essas características a tornam menos adequada para arquiteturas altamente paralelas e pipeline, essenciais no contexto de codificação de vídeo em tempo real. Para contornar essas limitações, optou-se pela utilização do método de Eliminação de Gauss-Jordan (ANDRILLI, 2023), que simplifica o fluxo de cálculo ao transformar diretamente a matriz em forma escalonada reduzida por linhas, eliminando a necessidade de retro-substituição e possibilitando uma implementação em hardware mais eficiente.

2. ARQUITETURA PROPOSTA

Este trabalho propõe uma arquitetura em hardware paralela e pipeline dedicada à resolução de Sistemas de Equações Lineares (LES), utilizando o método de Gauss-Jordan, e ao cálculo dos vetores de movimento incrementais (ΔMV) utilizados na predição affine do padrão Versatile Video Coding (VVC). O objetivo principal é reduzir a complexidade computacional desse processo e viabilizar sua execução em tempo real, mantendo alto desempenho e baixo consumo de energia, requisitos essenciais em sistemas embarcados e dispositivos com restrição de potência.

A arquitetura é mostrada na Figura 2, e é dividida em três estágios principais. O primeiro estágio, denominado Float Converter, realiza a conversão da matriz de entrada de 47 bits em valores de ponto flutuante de 32 bits, operação necessária devido às divisões envolvidas no processo de resolução dos sistemas lineares. Apesar de a matriz de entrada conter 42 elementos, a simetria da matriz de entrada permite reduzir o número de conversões, sendo necessárias apenas 27 no modelo affine de 6 parâmetros e 14 no modelo de 4 parâmetros. Todas essas conversões são realizadas em um único ciclo de clock, utilizando unidades de conversão implementadas por meio de pacotes de ponto flutuante em VHDL.

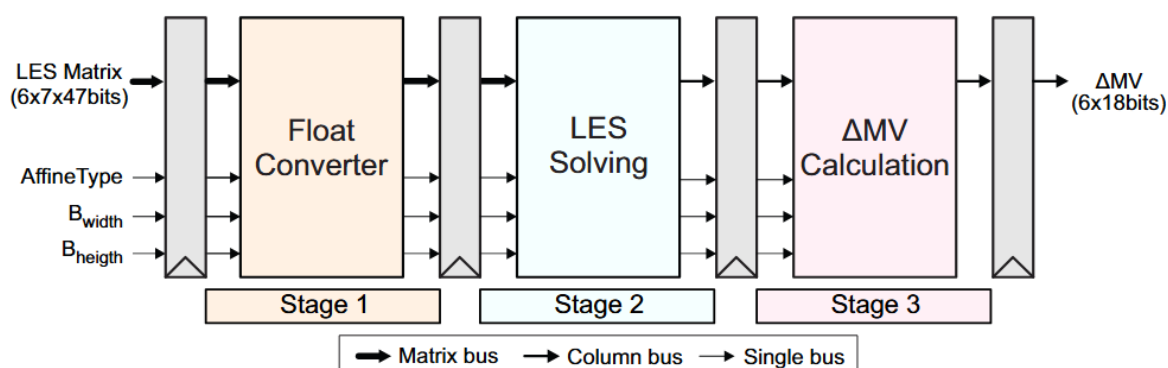


Figura 2: Arquitetura proposta

O segundo estágio, denominado LES Solving, é responsável pela resolução efetiva dos sistemas lineares. Para isso, são empregados até sete módulos paralelos que implementam o método da Eliminação de Gauss-Jordan, permitindo a solução simultânea de sete sistemas. Cada módulo é composto por um buffer de matriz, que armazena os dados em registradores de 32 bits, e um gerador de novas colunas, que utiliza um divisor, cinco multiplicadores e cinco subtratores. Esse arranjo possibilita atualizar colunas da matriz a cada iteração do algoritmo, reduzindo significativamente a latência. O resultado desse estágio é um conjunto solução que descreve as variações de movimento no bloco em processamento.

O terceiro e último estágio, denominado ΔMV Calculation, converte o conjunto solução nos valores finais de ΔMV que serão utilizados pela predição affine do VVC. Para maximizar a eficiência, o módulo é capaz de calcular um par de valores ΔMV em paralelo, aproveitando ao máximo o pipeline da arquitetura.

3. RESULTADOS E DISCUSSÃO

A arquitetura proposta foi descrita em VHDL e validada comparando os valores de ΔMV gerados pelo hardware com os da implementação em software. O design foi sintetizado para ASIC usando a biblioteca padrão TSMC 40 nm no Cadence RTL Compiler, com frequência de operação de 243 MHz, suficiente para processar 1,35 milhão de sistemas lineares por quadro de vídeo UHD 4K a 60 fps, entregando um resultado a cada três ciclos de clock. Os resultados de síntese, mostrados na Tabela I, mostram que o módulo LES Solving, com sete módulos de

Tabela I: Resultados

Module	Cell Area (kgates)	Total Power (mW)
Float Conversion	34.21	10.30
LES Solving	509.33	34.74
Gauss-Jordan Elimination	71.29	4.49
Matrix Buffer	7.94	1.30
New-Column Calculator	50.66	1.93
ΔMV Calculation	26.52	2.01
Total	570.06	47.32

Eliminação de Gauss-Jordan, possui o maior consumo de área e potência, principalmente devido ao New-Column Calculator. O módulo Float Converter apresenta área moderada, mas dissipação de potência relativamente alta, enquanto o módulo ΔMV Calculation tem baixo consumo de área e energia. No total, a arquitetura ocupa 570,06 kGates e dissipa 47,32 mW.

4. CONCLUSÕES

Foi apresentada uma arquitetura de hardware dedicada, em pipeline e paralela, para a resolução de sistemas lineares (LES) e o cálculo dos valores ΔMV na predição affine do padrão VVC. A arquitetura emprega o método de eliminação de Gauss-Jordan e consegue resolver um LES a cada três ciclos de clock. A síntese ASIC, direcionada para vídeos UHD 4K a 60 fps, apresenta uma dissipação de potência total de apenas 47,32 mW, evidenciando alta eficiência energética. A combinação de paralelismo e pipeline em todos os estágios permite que a

arquitetura proposta atinja um equilíbrio entre desempenho e eficiência energética superior às soluções anteriores encontradas na literatura. Enquanto trabalhos prévios apresentavam limitações em throughput, consumo de potência ou suporte ao cálculo completo de ΔMV , a abordagem aqui descrita supera esses desafios, entregando resultados em menor número de ciclos de clock e com menor dissipação de potência. Dessa forma, a proposta se mostra adequada para aplicações de codificação de vídeo em tempo real, incluindo cenários que exigem alta resolução e operação em dispositivos com restrições energéticas, como dispositivos móveis e sistemas embarcados.

5. REFERÊNCIAS BIBLIOGRÁFICAS

MUÑOZ, M. et al. Efficient Hardware Design for the VVC Affine Motion Compensation Exploiting Multiple Constant Multiplication. **2023 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)**, p.1-6, 2023.

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.

RODRIGUES DA SILVA, N.; PEREIRA DE SOUZA, F.; ROMANINI, E. **MÉTODO DE FATORAÇÃO LU PARA SOLUÇÃO DE SISTEMAS LINEARES**. Colloquium Exactarum. ISSN: 2178-8332, [S. l.], v. 9, n. 4, p. 41–47, 2018. Disponível em: <https://revistas.unoeste.br/index.php/ce/article/view/2267>. Acesso em: 22 set. 2023.

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.

MARTINA, M. **Simplified Affine Motion Estimation algorithm and architecture for the Versatile Video Coding standard**. 2022. Tese de Doutorado. Politecnico di Torino.

ANDRILLI, S.; HECKER, D. **Systems of Linear Equations. Elementary Linear Algebra**, p.57–100, 2023. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/B9780128229781000122>. Acesso em: 29 ago. 2025.