PROJETO DE HARDWARE PARA A ETAPA DE PRÉ-PROCESSAMENTO NA CODIFICAÇÃO DE VÍDEOS HDR NO PADRÃO HEVC

JOÃO BARTH; ALEX BORGES; BRUNO ZATT; LUCIANO AGOSTINI; MARCELO PORTO

Universidade Federal de Pelotas - Grupo de Arquiteturas e Circuitos Integrados
{jhpbarth, amborges, zatt, agostini, porto}@inf.ufpel.edu.br

1. INTRODUÇÃO

Os dispositivos de reprodução e os softwares de codificação atuais não possuem suporte para vídeos em HDR, este conteúdo deve ter sua representação transformada para se tornar codificável pelos padrões atuais. O tipo mais difundido de reprodução e armazenamento de arquivos é baseado em pixels com amostras de tamanho de 8 bits, com tipo de dados inteiro. Vídeos em HDR são representados utilizando um tipo especial de ponto flutuante, o *half-float*(IEEE, 1985), que consiste em pixels compostos de 3 amostras, contendo 16 bits cada. Este tipo de dado traz uma maior precisão no contraste e cor para o conteúdo de vídeo, entretanto estes benefícios vêm acompanhados por um grande acréscimo de armazenamento e de custo computacional para a codificação deste tipo de dado.

O padrão de codificação estado da arte a área de vídeo coding, o *High Efficiency Video Coding*(ITU-T, 2013), define que vídeos em HDR devem ser pré-processados antes de serem codificados, utilizando softwares padrão fornecidos pelo grupo. Este processo é composto por dois softwares (MANDEL et. al., 2014). O primeiro é o CTLRender, que transforma os pixels para que eles possam ser transformados de tipo sem uma perda muito expressiva em qualidade. O segundo é o tiff2ydzdx, que adapta o vídeo para um formato compatível com o HEVC.

Para estabelecer uma conversão que não resulte em uma grande perda na qualidade dos vídeos em HDR, este processo deve ser realizado utilizando diversas etapas que atuam corrigindo os valores de cor, contraste e saturação dos pixels para então ser realizada a conversão de tipo dos dados.

O pré-processamento de vídeos em HDR trabalha pixel por pixel, sem interdependência entre eles, deixando este processo extremamente paralelizável, podendo trabalhar com diversos pixels simultaneamente. Entretanto, operadores e processos com ponto flutuante em hardware são extremamente custosos e lentos, limitando a eficiência da implementação em hardware sem otimizações focadas em melhoria de desempenho e redução de área.

O processo de conversão, além de paralelizável, é um tanto quanto linear, proporcionando a introdução de pipelines entre as etapas, aumentando a eficiência do processo. Uma proposta de otimização para esta arquitetura é apresentada em um trabalho prévio do grupo (BORGES et. al., 2014).

Neste trabalho será demonstrado o fluxo de dados do pré-processamento de vídeos em HDR, os desafios agregados com a implementação em hardware de arquiteturas em ponto flutuante e discutiremos resultados parciais de síntese de hardware, focando os blocos mais custosos em área.

2. METODOLOGIA

A implementação de uma arquitetura de hardware que realiza o pré-processamento de vídeos em HDR está sendo descrita em linguagem VHDL,

sendo utilizada a ferramenta Quartus II e o dispositivo FPGA 5SGXMBBR3H43I4, da família Stratix V da Altera (REF). Além disso, está sendo utilizada uma biblioteca específica de funções para uso de ponto flutuante (BISHOP, 2010). Além desta biblioteca, também foram implementados operadores que eram inexistentes na biblioteca utilizada. Os operadores consistem em blocos que calculam séries, foram inclusos os cálculos de arco tangente, logaritmo de base natural e também o cálculo de potência com números fracionários como expoente.

O projeto desta arquitetura de hardware foi focado em representar em hardware o processo de pré-processamento realizado pelo software CTLRender (MANDEL et. al., 2014). Esta implementação tem como objetivo projetar um hardware dedicado para a etapa de pré-processamento na codificação de vídeos HDR, demonstrando a possibilidade de implementação deste processo em hardware, destacando os gargalos deste processo e ressaltando a necessidade de paralelização do processo, identificando onde otimizações são mais pertinentes. O fluxo de dados da arquitetura em questão é demonstrado na Figura 1, descrevendo as diferentes etapas da transformação das amostras, dividindo-as por blocos.

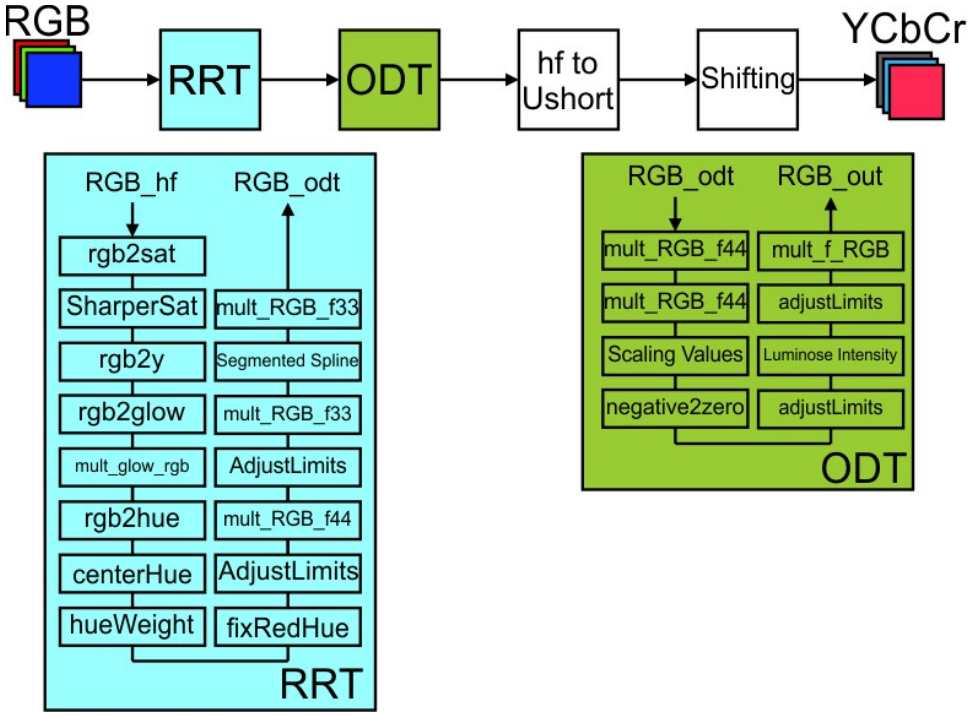

Figura 1 – Fluxo de pré-processamento.

A etapa RRT recebe um pixel composto de três amostras de 16 bits em ponto flutuante e tem por objetivo achar propriedades dos pixels, e realizar ajustes tendo estas características como base. “rgb2sat” e “ShaperSat” tem por objetivo corrigir a saturação, ”rgb2y”, “mult_glow_rgb” e rgb2Glow” tem por objetivo ajustar o Brilho,”rgb2hue”, “centerHue”, “hueWeight e fixRedHue” corrigem a matiz dos pixels. blocos com o nome iniciado por “mult“ realizam multiplicações de matrizes utilizando dados calculados anteriormente. Os blocos denominados “*AdjustLimits”* realizam testes para manter as amostras dentro do range máximo e mínimo. As etapas “*rgb2hue”* e “Segmented Spline” são as mais complexas, tendo em vista que elas utilizam fórmulas iterativas de complexidade quadrática entre seus cálculos. O bloco “Segmented Spline” define um novo valor par ao pixel dependendo de seu valor na curva “spline”.

Neste trabalho são apresentados os resultados preliminares para o desenvolvimento do módulo RRT, mais especificamente para os módulos “Segmented Spline” e “rgb2hue” que são só mais complexos deste fluxo, contendo cálculos complexos.

Na execução desta arquitetura, os desafios são apresentados na forma da implementação dos operadores complexos presentes no trabalho, entre eles, estão blocos que resolvem fórmulas como, arco tangente, logaritmo natural e potência elevada a números fracionários. Estes blocos são dependentes de séries, sendo que o cálculo de seus valores deve ser iterado diversas vezes para que o resultado seja preciso. Tendo em vista a síntese de hardware, o uso de séries pode causar um grande gargalo tanto em questões de área, quanto em questão de frequência, dado que o processo de pré-processamento deve ser extremamente preciso.

Para implementar as séries, presentes nos módulos “Segmented Spline” e “rgb2hue” foram projetados blocos em hardware que realizam os cálculos com um número fixo de iterações. Implementações utilizando iterações não são as mais adequadas para uma adaptação em hardware. Estas etapas podem ser otimizadas utilizando diferentes abordagens, dentre elas está a aproximação utilizando retas, dado que tais séries formam ondas que podem ser discretizadas em retas, tomando ranges diferentes para cada reta.

Tendo em vista um vídeo em resolução full HD, contendo 1920x1080 pixels e 30 frames por segundo, seria preciso uma arquitetura em hardware trabalhando com uma frequência de operação de no mínimo 62.3MHz para obter um sistema em tempo real.

O desenvolvimento da arquitetura do pré-processador de vídeos HDR ainda está em fase de desenvolvimento, apenas os blocos “Segmented Spline” e RGB2Hue” estão prontos e validados.

3. RESULTADOS E DISCUSSÃO

Os resultados obtidos com a síntese em FPGA dos módulos “Segmented Spline” e “rgb2hue” são apresentados na Tabela 1, tendo sua performance medida em duas resoluções diferentes, Full HD(1920x1080) e CIF(352x288). Estes blocos são os mais complexos deste processo, nestes blocos estão inclusas as fórmulas dependentes de aproximações baseadas em séries.

Tabela 1 – Resultados de síntese em FPGA para o módulo RRT

	Segmented Spline	Rgb2Hue
Área (ALUTs)	6262	7872
Frequência(MHz)	152.3	4.2
Pinos	32	65
Registradores	11	32
Quadros por segundo(Full HD)	73	2
Quadros por segundo(CIF)	1502	41

Estes blocos devem ser integrados em uma versão final da arquitetura, logo resultados parciais demonstram a inaptidão destes blocos de atingir uma frequência aceitável de operação. Apesar do bloco “Segmented Spline” atingir uma frequência de operação aceitável para vídeos em Full HD, o bloco “rgb2hue” não consegue atingir a taxa de processamento mínima de 30fps. A junção destes blocos com o restante da arquitetura resultaria na soma dos atrasos das etapas, diminuindo a frequência de operação.

O bloco “rgb2hue” possui séries para o cálculo do arco tangente no seu processo de extração do matiz do pixel, este processo é custoso pois ele realiza diversas iterações para obter um resultado preciso. Este processo deve ser otimizado para aumentar a frequência de operação da arquitetura de hardware, isto pode ser atingido calculando diversos passos da iteração simultaneamente e realizando a soma dos resultados no final do processo.

O foco do desenvolvimento desta arquitetura é atingir uma taxa de processamento capaz de suprir 30 frames por segundo para um vídeo em resolução full HD. A integração destes módulos se prova incapaz pela taxa de saída do bloco “rgb2hue”, que por si só limita a taxa de saída para dois frames por segundo.

4. CONCLUSÕES

Os resultados preliminares apresentados neste trabalho mostram que o pré-processamento de vídeos em HDR possui etapas custosas e lentas e, que para uma implementação eficiente em hardware, seria necessário tomar medidas para otimizar o processo.

Para obtermos uma arquitetura em hardware com desempenho suficiente para o processamento de vídeo de alta resolução em tempo real, outras técnicas devem ser aplicadas, como a utilização de pipeline, paralelização de amostras e o uso de outros tipos de dados, como o ponto fixo.

Os trabalhos futuros consistem em finalizar a implementação completa da arquitetura e, então, validar e gerar os resultados de síntese para FPGA. Após a conclusão da primeira versão do pré-processador, uma versão otimizada da arquitetura será implementada, visando atingir pré-processamento em tempo real de vídeos em formato HDR com resolução HD 1080p (1920x1080 pixels).

5. REFERÊNCIAS BIBLIOGRÁFICAS

IEEE. **IEEE Standard for Binary Floating-Point Arithmetic**. The institute of Eletrical and Eletronics Engineers, New York, 1985. Acessado em 21 ago. 2015. Online. Disponível em: http://www.zh.com.br/especial/index.htm

ITU-T. International Telecomunication Union. **Recommendation ITU-T H.265: High Efficiency Video Coding**. Abril, 2013.

MANDEL, B.; FOGG, C.; HELMAN, J. High Dynamic Range / Wide Color Gamut workflow. **JCT-VC Meeting Valencia**, Valencia, Abril 2014. Acessado em 21 ago. 2015. Online. Disponível em: http://phenix.int-evry.fr/jct/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q0085-v3.zip

BISHOP, DAVID. EDA Industry Working Groups. **Floating Package User’s Guide**. Acessado em 21 ago. 2015. Online. Disponível em: http://www.eda.org/fphdl/Float_ug.pdf

BORGES, A.; BARTH, J.; ZATT, B.; AGOSTINI, L.; PORTO, M. Hardware Architecture Proposal for the High Dynamic Range Video Pre-Processing. In: SOUTH MICRO-ELECTRONICS SYMPOSIUM, 30, Santa Maria, 2015. **Anais...** Santa Maria: EMICRO/SIM, 2015.