ARQUITETURA DE HARDWARE DEDICADA PARA A QUANTIZAÇÃO DO PADRÃO HEVC SEM O USO DE MULTIPLICADORES

LUCIANO BRAATZ; LUCIANO AGOSTINI; BRUNO ZATT; MARCELO PORTO

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

1. INTRODUÇÃO

O codificador de vídeo de alta eficiência (HEVC, do inglês *High Efficiency Video Coder*) é o estado da arte em codificação de vídeo e permite uma redução de aproximadamente 50% na taxa de bits em relação aos padrões de codificação anteriores, para uma qualidade perceptual equivalente, conforme SULLIVAN et al. (2012).

Segundo UGUR (2010), essa eficiência de codificação é devido à adoção de vários novos métodos de codificação, que inclui um esquema de quantização altamente otimizado. A quantização e quantização inversa estão fortemente integradas na cadeia de processamento do HEVC, sendo utilizadas, no mínimo, duas vezes no laço de transformadas. Esse uso intenso faz com que o desempenho computacional do HEVC dependa bastante da latência e taxa de transferência atingida pelos circuitos de quantização e quantização inversa.

Segundo SZE; BUDAGAVI; SULLIVAN (2014), a quantização consiste na divisão dos coeficientes transformados pelo passo de quantização ($QStep$) e subsequente arredondamento, que reduz a maioria dos coeficientes quantizados a zero. O parâmetro de quantização (QP) define o valor de $QStep$, conforme (1). QP pode assumir valores na faixa de 0 a 51 e foi especificado de forma que $QStep$ dobra a cada incremento de 6 no valor de QP.

$$QStep = \left(2^{1/6}\right)^{QP-4} \tag{1}$$

O valor de $QStep$ é fracionário, que exige um grande esforço computacional para seu cálculo. O esforço pode ser reduzido através do calculo prévio das potências fracionárias e ajuste dos cálculos para ponto fixo, conforme (2), onde $g = [40,45,51,57,64,72]^T$ e $QP \bmod 6$ é o resto da divisão de QP por 6.

$$QStep(QP) = g\left(QP \bmod 6\right) << \frac{QP}{6} - 6 \tag{2}$$

A equação acima simplifica o cálculo de $QStep$, mas a quantização ainda sofre com o custo computacional de uma divisão. Como os valores dos divisores são conhecidos, o software de referência HEVC Test Model (JCT-VC, 2015) armazena a divisão de 2^{19} por esses valores e efetua uma multiplicação pelos valores armazenados, corrigindo com um deslocamento de 19 bits à direita. A expressão utilizada no software de referência é dada por (3), onde Z é o coeficiente quantizado, W é o coeficiente da transformada e $QStep'$ é o valor armazenado.

$$Z = sign(W)\left[\left(|W| \times QStep' + offset\right) >> qBits\right] \tag{3}$$

O termo $qBits$, expresso em (4), efetua a correção discutida acima e normaliza os coeficientes em relação ao tamanho do bloco. O símbolo $\lfloor\ \rfloor$ representa um arredondamento para baixo e $TrSize$ é o logaritmo de base 2 do tamanho do bloco.

$$qBits = 21 + \left\lfloor \frac{QP}{6} \right\rfloor - TrSize \tag{4}$$

O termo *offset*, expresso em (5), corrige os erros de arredondamento e depende do tipo de predição utilizada e do parâmetro $qBits$.

$$offset = \begin{cases} 171 << (qBits - 9), se\ slice\ type\ I \\ 85 << (qBits - 9), caso\ contrário \end{cases} \tag{5}$$

O HEVC suporta quantização dependente da frequência, utilizando matrizes de quantização diferentes para cada tamanho de bloco e cada resto da divisão de QP por 6. Conforme GONÇALVES (2014), uma análise dos coeficientes mostrou uma relação entre as matrizes de tamanhos diferentes, onde a matriz 8x8 armazena todas as informações necessárias para representar as matrizes 4x4, 16x16 e 32x32.

2. METODOLOGIA

A operação mais critica de (3) é a multiplicação entre dois termos de 16 bits. O método normal para efetuar uma multiplicação binária é através da soma deslocada dos produtos do multiplicando por cada bit do multiplicador. Os valores utilizados como multiplicador são conhecidos, assim a operação foi codificada através de somas e deslocamentos estritamente necessários para a execução da multiplicação. A comparação das diferentes matrizes mostra a existência de uma relação de proporção entre elas. Portanto, a multiplicação foi efetuada pelo primeiro coeficiente da matriz, conforme QP, e ajuste da escala para o valor correto, conforme a posição do coeficiente na matriz. O ajuste tomou como base a matriz cujo primeiro coeficiente é 16384, por ser base de dois (16384 = 2^{14}), assim o resultado da segunda multiplicação é deslocado 14 bits à direita.

A arquitetura proposta é apresentada na Figura 1, onde o bloco *SeparaQP* gera o quociente e o resto da divisão de *QP* por 6 e os blocos *Tabela4x4* e *Tabela8x8* efetuam a multiplicação por *QStep'*, através da multiplicação pelo primeiro coeficiente da matriz e o ajuste à escala para o valor correto, respectivamente.

Detalhes do bloco Tabela4x4 são apresentados na Figura 2. O uso racional dos recursos se dá pela reutilização de resultados e através do balanceamento dos atrasos na composição das somas.

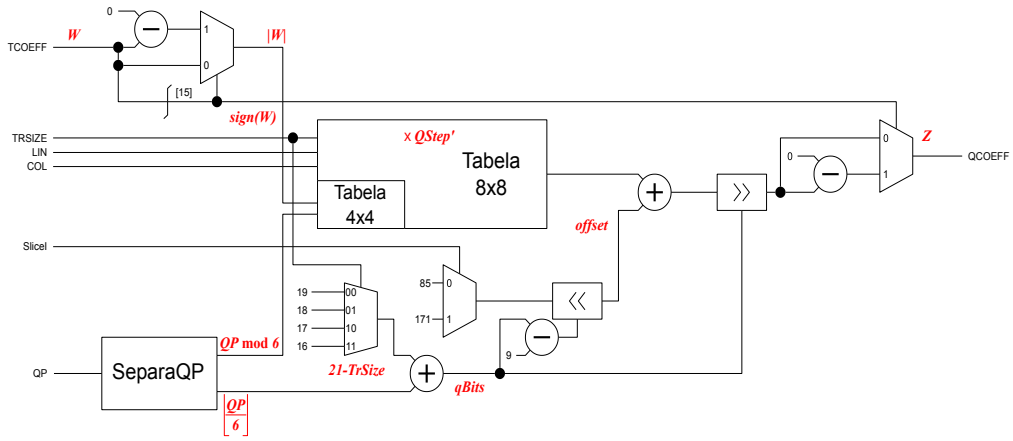

Figura 1 – Arquitetura proposta para o Quantizador

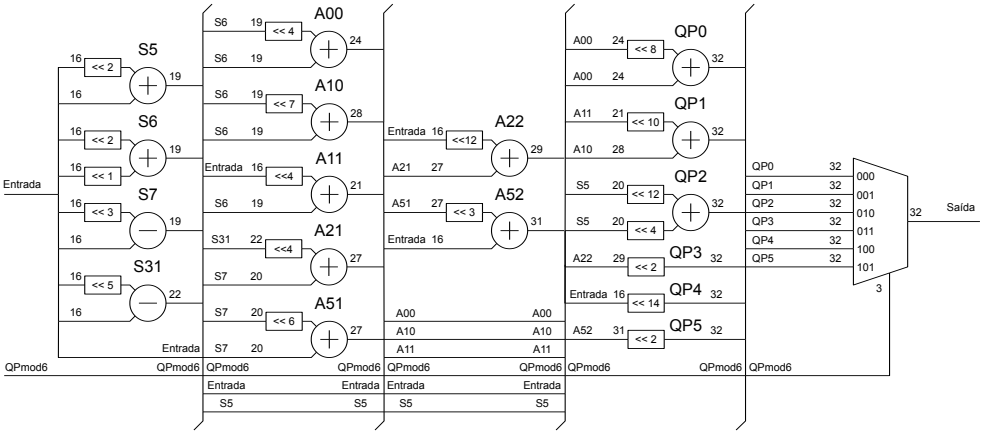

Figura 2 – Diagrama do bloco Tabela 4x4

O resultado da soma *S6*, por exemplo, é utilizada cinco vezes no segundo nível de cálculos e balanceamento dos atrasos ocorre, por exemplo, na saída *QP0*, conforme descrito de (6) a (8), onde $<< n$ representa deslocamento de n bits à esquerda.

$$QP0 = Entrada \times 0110.0110.0110.0110 \quad (6)$$

$$QP0 = S6 << 12 + S6 << 8 + S6 << 4 + S6, onde\ S6 = Entrada << 2 + Entrada << 1 \quad (7)$$

$$QP0 = A00 << 8 + A00, onde\ A00 = S6 << 4 + S6 \quad (8)$$

A implementação da arquitetura do quantizador foi realizada através da linguagem de descrição de hardware VHDL, para implementação em dispositivo FPGA (*Field-Programmable Gate Array*), utilizando o software Quartus II, da Altera. A validação foi baseada em dados extraídos do software de referência, onde o vídeo *Foreman* (XIPH, 2016) foi codificado e os dados de entrada e saída do quantizador foram salvos em um arquivo texto. A codificação foi feita para dois quadros do vídeo, configurado para o perfil main e QPs de 22, 27, 32 e 37. A validação foi efetuada no software ModelSim-Altera 10.1d, utilizando 591331 conjuntos de dados, onde as entradas foram aplicadas à arquitetura e sua saída foi comparada com a saída extraída do software de referência.

3. RESULTADOS E DISCUSSÃO

A arquitetura desenvolvida utiliza 1464 ALMs e 2519 registradores do FPGA Stratix V 5SGXEA7N1F45C2L, sem utilizar DSPs (processadores digitais de sinais), e opera à frequência máxima de 361,27 MHz. A arquitetura é capaz de processar até 116,15 quadros por segundo, muito mais do que os 30 quadros por segundo utilizados normalmente em vídeos, portanto a arquitetura atual permite processamento de vídeos em tempo real.

A Tabela 1 apresenta uma comparação entre os resultados de síntese obtidos na arquitetura desenvolvida e um trabalho relacionado.

Tabela 1 – Comparação dos Resultados

	Arquitetura Proposta	**GONÇALVES (2014)**
ALMs	1464	4154
Registradores	2519	1740
DSPs	0	32
Amostras / ciclo	1	32
Frequência Máxima (MHz)	361,27	148,77
Desempenho (fps HD 1080p)	116,15	1530,50

A arquitetura desenvolvida atinge uma frequência máxima de processamento 2,42 vezes maior, utiliza cerca de um terço da quantidade de ALMs, entretanto o desempenho é 13 vezes menor por se tratar de uma arquitetura que não explora paralelismo, já que processa uma amostra por ciclo.

A diferença na utilização de recursos do FPGA se deve ao uso de 32 blocos DSP por GONÇALVES (2014) para efetuar as multiplicações, que reduz a quantidade de recursos do FPGA (ALMs e registradores) e aumenta o desempenho da solução. Segundo ALTERA (2016), os blocos DSP apresentam alta eficiência energética e operam a frequências muito maiores que os circuitos equivalentes sintetizados nos FPGAs. Entretanto a solução adotada na arquitetura proposta, sintetizada totalmente em FPGA, permite a operação em frequência mais alta que ocorre em GONÇALVES (2014).

A utilização de DSPs por GONÇALVES (2014) reduz significativamente o custo de hardware da arquitetura. Considerando o FPGA utilizado, é estimado que um DSP utilize até 329 ALMs. Com isso, a arquitetura apresentada em GONÇALVES (2014) utilizaria cerca de 10 vezes mais ALMs que a arquitetura desenvolvida, caso não utilizasse DSPs.

4. CONCLUSÕES

Este artigo apresenta uma abordagem para efetuar a quantização no padrão HEVC que não utiliza multiplicadores binários, reduzindo a quantidade de somas necessárias para efetuar a multiplicação binária necessária ao quantizador, com reutilização de somas e balanceamento das somas para reduzir o caminho crítico.

A sequência deste trabalho tratará da implementação com a multiplicação efetuada em somente um nível, para comparação do número de recursos utilizados no FPGA e da frequência máxima. Após a comparação, a arquitetura mais adequada será paralelizada para permitir o processamento de 32 amostras simultâneas.

5. REFERÊNCIAS BIBLIOGRÁFICAS

SZE, V.; BUDAGAVI, M; SULLIVAN, G. **High Efficiency Video Coding (HEVC) – Algorithms and Architectures**. Switzerland: Springer, 2014.

SULLIVAN, G. et al. Overview of the High Efficiency Video Coding (HEVC) Standard. **IEEE Trans. Circuits Syst. Video Techonol.**, v.22, n.12, p. 1649–1668. 2012

GONÇALVES, G. W. **Desenvolvimento de um IP core para DCT e Quantização segundo padrão HEVC: Projeto em Electronic System Level**. 2014. 62f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação) - Universidade Federal de Pelotas.

JCT-VC. **Subversion repository for the HEVC test model reference software** Acessado em 22 nov. 2015. Online. Disponível em: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware

XIPH. **Xiph.org – Derf's Test Media Collection**. Acessado em 15 mai. 2016. Online. Disponível em: https://media.xiph.org/video/derf

ALTERA. **Stratix: Digital Signal Processing Blocks**. Acessado em 20 jul. 2016. Online. Disponível em: https://www.altera.com/products/fpga/features/stx-dsp-block.html