SOLUÇÃO AUTOMATIZADA PARA MINIMIZAR O CUSTO DE HARDWARE DA DCT DO PADRÃO HEVC ATRAVÉS DO COMPARTILHAMENTO DE SUB-EXPRESSÕES E ELIMINANDO O USO DE MULTIPLICAÇÕES

CONCEIÇÃO, Ruhan[1]; JESKE, Ricardo[2]; AGOSTINI, Luciano[3]; MATTOS, Júlio[3]

[1]Universidade Federal de Pelotas, Curso de Engenharia de Computação, PET Computação
[2]Universidade Federal de Pelotas, Programa de Pós-Graduação em Computação
[3]Universidade Federal de Pelotas, Centro de Desenvolvimento Tecnológico
{radconceicao,rgjeske,agostini,julius}@inf.ufpel.edu.br

1 INTRODUÇÃO

Através dos últimos anos, dispositivos capazes de reproduzir e/ou capturar vídeos digitais em alta definição estão sendo cada vez mais utilizados. Pode-se notar este fato através da evolução no uso de *smartphones*, *tablets*, televisores, etc. Quando não comprimidos, estes vídeos necessitam de um expressivo volume de dados para serem armazenados (Agostini, 2007). Desta forma, a codificação de vídeos é uma técnica muito importante para viabilizar o armazenamento e a transmissão dos mesmos.

Atualmente, o estado da arte em codificação de vídeos é o padrão H.264/AVC. Este padrão foi lançado em 2003, atingindo resultados superiores em codificação de vídeos, quando comparado aos demais padrões, como, por exemplo, o padrão MPEG-2 (Agostini, 2007). A partir de janeiro de 2010, um grupo denominado JCT-VC (JCT-VC, 2012) iniciou o desenvolvimento de um novo padrão em codificação de vídeos, o padrão de alta eficiência em codificação de vídeos (*High Efficient Video Coding* - HEVC) (HEVC, 2011), cujo lançamento está previsto para 2013.

Entre as principais metas do HEVC, destacam-se atingir o dobro na taxa compressão – comparado ao H.264/AVC – sem perda alguma na qualidade de imagem. O HEVC dará suporte à codificação de vídeos com resolução superior ao *Full HD* (resolução máxima suportada pelo H.264/AVC), como por exemplo, a resolução *Quad Full HD* (3840 X 2160 pixels).

O processo de codificação de vídeos é realizado através de uma sequência de passos, onde cada um destes passos tem um papel fundamental no processo de compressão. A figura 1 ilustra um codificador de vídeos genérico. O módulo T (módulo das transformadas), destacado na imagem, indica onde está o foco deste trabalho. Este módulo recebe como entrada matrizes residuais geradas pelos blocos de predição, sendo então transformadas para o domínio das frequências a fim de ampliar as taxas de compressão atingidas pelas próximas etapas do codificador (quantização e codificação de entropia).

A principal transformada utilizada no módulo T, e foco deste artigo, é a transformada discreta dos cossenos (DCT). O HEVC estipula quatro tamanhos de transformada: 4x4, 8x8, 16x16 e 32x32. Para realizar o cálculo de uma transformada 2-D, é necessário aplicar a transformada 1-D sobre cada linha da matriz de entrada, transpor o resultado e repetir o processo novamente.

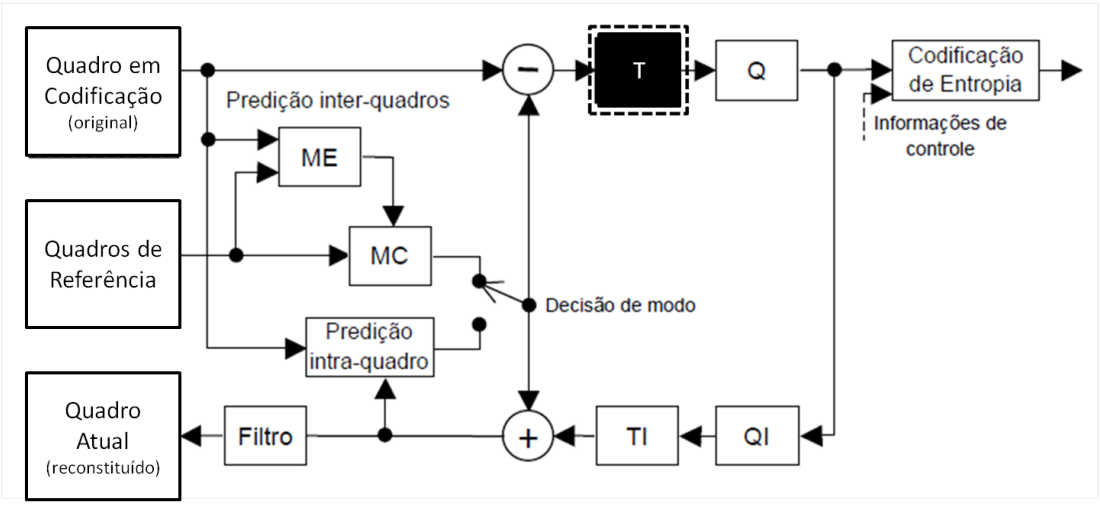

Figura 1 – Diagrama de blocos de um codificador de vídeos genérico

Para realizar o processo da transformada, é necessária uma grande quantidade de cálculos com as matrizes envolvidas no processo. Desta forma, o desenvolvimento de arquiteturas de hardware dedicadas para os codificadores de vídeos é uma alternativa importante para viabilizar esta tecnologia.

Algumas estratégias de otimização podem ser aplicadas para reduzir o número de cálculos necessários e/ou reduzir a complexidade destes cálculos. Muitas sub-expressões utilizadas no processo do cálculo da DCT podem ser compartilhadas entre as diversas equações, assim reduzindo o número cálculos. Os multiplicadores podem ser substituídos por somas e deslocamentos, diminuído o custo computacional das operações realizadas.

Este artigo apresenta o desenvolvimento de um software que utiliza algumas heurísticas para transformar as multiplicações em somas de deslocamentos e para avaliar as possibilidades de combinações entre os cálculos envolvidos na DCT na tentativa de maximizar os compartilhamentos de sub-expressões entre estes cálculos e, deste modo minimizar o custo do futuro hardware que será projetado. O foco inicial das otimizações está na transformada discreta dos cossenos 1-D de 32 pontos definida pelo HEVC. O software avalia cerca de cinco bilhões de possibilidades, retornando aquelas menos custosas.

2 METODOLOGIA (MATERIAL E MÉTODOS)

Para o cálculo da DCT 1-D de tamanho 32 são utilizadas dezesseis constantes na parte abrangida pela otimização buscada em software. É importante destacar que o processo é aplicado apenas nas equações ímpares de saída, uma vez que as equações pares definem a DCT 1-D de tamanho 16 (Hecktheuer, 2012). A Tabela 1 mostra exemplos de equações utilizadas no processo do cálculo da DCT 1-D de tamanho 32. Como pode ser visto, as variáveis (a_1, a_3, etc.) são multiplicadas pelas 16 constantes dentro das equações. Uma vez que multiplicadores são custosos em termos de hardware, as multiplicações são substituídas por somas e deslocamentos.

O software desenvolvido que utiliza algumas heurísticas para transformar as multiplicações em somas de deslocamentos e para avaliar as possibilidades de combinações entre os cálculos envolvidos na DCT. O software foi dividido em sete diferentes módulos.

O primeiro módulo – módulo 1 - do software desenvolvido é responsável por avaliar as possibilidades de somas e deslocamentos para substituir as multiplicações. Por exemplo, na equação X_1, a variável a_{29} é multiplicada pela constante 13, mas essa mesma operação poderia ser substituída por várias formas de somas e deslocamentos como mostra a tabela 2.

Tabela 1 – Exemplos de equações utilizadas na DCT 1-D 32

X_n	Equação Final
X_1	$90*a_1 + 90*a_3 + 88*a_5 + 85*a_7 + 82*a_9 + 78*a_{11} +$ $+73*a_{13} + 67*a_{15} + 61*a_{17} + 54*a_{19} + 46*a_{21} +$ $+38*a_{23} + 31*a_{25} + 22*a_{27} + 13*a_{29} + 4*a_{31}$
...	...
X_{31}	$4*_{a1} - 13*_{a3} + 22*a_5 - 31*a_7 + 38*a_9 - 46*a_{11} +$ $+54*a_{13} + 61*a_{15} + 67*a_{17} - 73*a_{19} + 78*a_{21} +$ $-82*a_{23} + 85*a_{25} - 88*a_{27} + 90*a_{29} - 90*a_{31}$

Tabela 2 – Exemplo de somas e deslocamentos possíveis de serem utilizados ao invés de $a_{29} * 13$

Alguns Possíveis Deslocamentos	Possíveis Combinações para Gerar a Multiplicação $a_{29} * 13$
$a_{29}<<1 = a_{29} * 2$	$a_{29}<<3 + a_{29} <<2 + 1 = a_{29} * 13$
$a_{29} <<2 = a_{29} * 4$	$a_{29}<<3 + a_{29}<<2 + a_{29}<<1 – 1 = a_{29} * 13$
$a_{29}<<3 = a_{29} * 8$	$a_{29}<<4 - a_{29}<<3 + a_{29}<<2 + 1 = a_{29} * 13$
$a_{29} <<4 = a_{29} * 16$	$a_{29}<<4 - a_{29}<<2 + a_{29}<<1 + 1 = a_{29} * 13$
$a_{29}<<5 = a_{29} * 32$	$a_{29}<<4 - a_{29}<<2 + a_{29}<<1 - 1 = a_{29} * 13$
$a_{29}<<6 = a_{29} * 64$	$a_{29}<<4 - a_{29}<< 2 - 1 = a_{29} * 13$
$a_{29}<<7 = a_{29} * 128$	$a_{29}<<5 - a_{29}<<4 - a_{29}<<2 + 1 = a_{29} * 13$
...	$a_{29}<<5 - a_{29}<<4 - a_{29}<<1 - 1 = a_{29} * 13$

O módulo seguinte – módulo 2 – é responsável por combinar todas as possibilidades geradas para cada uma das dezesseis constantes com as demais somas previstas nas equações. Ao se multiplicar o número total de possibilidades gerada para cada uma das dezesseis constantes, podemos inferir que o módulo 2 gera cerca de cinco bilhões de combinações. Desta forma, o software nos módulos seguintes, deverá decidir, em qual dessas combinações se dará a melhor configuração em hardware.

Como pode ser visto na tabela 1, a ordem em que as constantes são utilizadas, tal como seus sinais, variam entre as dezesseis equações. O terceiro módulo – módulo 3 – é responsável por rearranjar as constantes (já substituídas por somas e deslocamento) entre as dezesseis equações. Além disso, esse módulo transpõe as matrizes geradas pelo módulo 2, ou seja, anteriormente, cada linha da matriz representava uma constante, depois do módulo 3, cada linha passa a representar o deslocamento efetuado. Além disso, cada coluna representa cada uma das variáveis (a_1, a_3, etc.).

O módulo 4 é responsável por, a partir da saída gerada pelo módulo 3, procurar operações para serem compartilhadas entre cada linha de cada uma das

dezesseis equações utilizadas no cálculo da DCT. Primeiramente o módulo procura qual operação (a_{n1} +/- a_{n2}) detém maior ocorrência. Após isto, a mesma é substituída por uma operação nomeada b_n em todas suas ocorrências no código. Esse processo é repetido até que sobre em cada linha, no máximo uma variável a_n.

Uma técnica alternativa empregada na busca é primeiramente verificar se existe apenas um par de variáveis em alguma das linhas de cada equação, e caso seja encontrado, a mesma é definida como uma operação b_n e todas suas ocorrências são alteradas dentro das equações.

Os módulos seguintes, 5, 6 e 7, seguem a mesma idéia do módulo 4, porém os mesmos criam as operações *c's, d's* e *e's* respectivamente.

3 RESULTADOS E DISCUSSÃO

Até o presente momento, o software ainda não finalizou sua execução (pois existem mais de 5 bilhões de possibilidades). Mas alguns resultados puderam ser coletados. Quando realizada a otimização de forma manual, foram encontrados, no melhor caso, 106 somas *b's*, enquanto o software até o presente momento de execução, encontrou uma solução que utiliza apenas 37 somas de *b's*.

Pode-se perceber, com isso, que o software está cumprindo as metas iniciais, que era encontrar um hardware mais otimizado para o cálculo das transformadas discretas dos cossenos.

4 CONCLUSÃO

Este artigo apresentou um software que busca, entre cerca de cinco bilhões de possibilidades, a solução mais otimizada para o futuro projeto de hardware da transformada discreta dos cossenos 1-D de 32 pontos do HEVC. A otimização é obtida através da aplicação de heurísticas que reduzem o número de somas ou subtrações e eliminam as multiplicações usadas nos cálculos.

Resultados iniciais mostram que o software está atingindo as metas iniciais propostas, mesmo que sua execução não tenha ainda sido finalizada.

5 REFERÊNCIAS

AGOSTINI, Luciano. **Desenvolvimento de Arquiteturas de Alto Desempenho Dedicadas a Compressão de Vídeo Segundo o Padrão H.264/AVC**. Tese (Doutorado em Ciência da Computação). Porto Alegre: Instituto de Informática, UFRGS, 2007.

HECKTHEUER, Bruno; CONCEIÇÃO, Ruhan; SOUZA, J.C; JESKE, Ricardo; AGOSTINI, Luciano; MATTOS, J.C.B. Reconfigurable Architecture Implementation for the 1-D Discrete Cosine Transform. In: **Southern Programmable Logic Design Forum**, 2012, Proceedings ... Pelotas: UFPel, 2012.

Joint Collaborative Team on Video Coding (JCT-VC). Disponível em: <http://www.itu.int/en/ITU-T/studygroups/com16/video/Pages/jctvc.aspx>. Acesso em: 01 julho 2012.

ISO/IEC-JTC1/SC29/WG11. **HEVC Reference Software Manual**. Geneva, Suíça, 2011.