



DESENVOLVIMENTO DE ARQUITETURAS MULTITRANSFORMADAS DE ELEVADA TAXA DE PROCESSAMENTO PARA O PADRÃO H.264/AVC

CORRÊA, Marcel; THUROW, Mateus; SAMPAIO, Felipe; AGOSTINI, Luciano

*Grupo de Arquiteturas e Circuitos Integrados (GACI) – IFM/DINFO/UFPel
Campus Universitário – Caixa Postal 354 – CEP 96010-900. mmc Correa.ifm@ufpel.edu.br*

1. INTRODUÇÃO

A compressão de vídeos é uma técnica utilizada para tornar possível o armazenamento e a transmissão de vídeos digitais de alta qualidade os quais, sem este processo de compressão, utilizariam muitos bits para serem representados. O padrão H.264/AVC (ITU-T, 2003) é o mais novo padrão de compressão de vídeo definido por especialistas da ITU-T e ISO/IEC com o intuito de obter taxas de compressão duas vezes maiores que os padrões já existentes como MPEG-2. Esse padrão também é o que possui maior complexidade computacional entre todos os padrões existentes.

Devido a esta alta complexidade computacional, soluções para compressores H.264/AVC que atinjam processamento de tempo real (30 quadros por segundo) para vídeos de alta resolução se tornam inviáveis por meio de software, incentivando o desenvolvimento de soluções em hardware.

O H.264/AVC é o padrão adotado pelo Sistema Brasileiro de TV Digital (SBTVD) e, para sua implementação, diversos pesquisadores estão estudando soluções arquiteturais para compressores H.264/AVC.

Os principais módulos do codificador H.264/AVC são: predição inter-quadros (composto pelos módulos de estimação de movimento – ME – e compensação de movimento – MC), predição intra-quadros, transformadas diretas (T), e inversas (IT), quantização direta (Q) e inversa (IQ), filtro redutor do efeito de bloco e codificador de entropia (RICHARDSON, 2003). A Fig. 1 mostra o diagrama em blocos do compressor H.264/AVC.

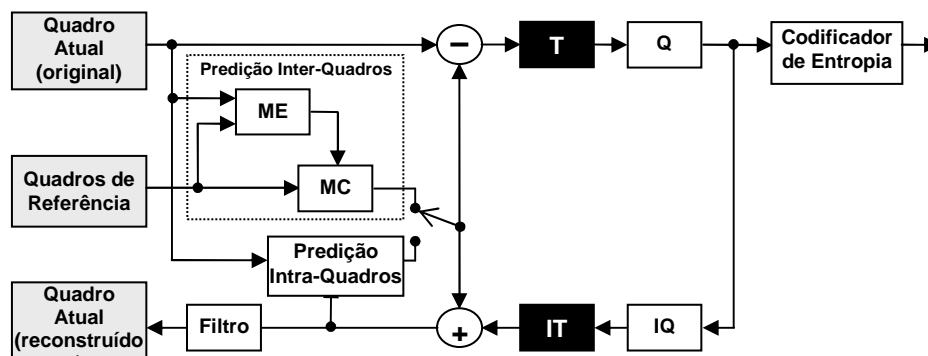


Figura 1. Diagrama de blocos do codificador H.264/AVC.

Os blocos destacados em preto na Figura 1 são os módulos das transformadas diretas e inversas, os quais são o foco deste trabalho.

A função das transformadas é transformar blocos 4x4 de resíduos, gerados pela etapa de predição inter ou intra, do domínio espacial para o domínio das frequências, para que o módulo de quantização possa mais facilmente manipulá-los.

O padrão define quatro diferentes transformadas para blocos 4x4 que são: a transformada discreta dos cossenos 4x4 direta (FDCT 4x4), transformada discreta dos cossenos 4x4 inversa (IDCT 4x4), transformada Hadamard 4x4 direta (FHAD 4x4) e transformada Hadamard 4x4 inversa (IHAD 4x4). Estão presentes também duas diferentes transformadas para blocos 2x2, que são a transformada Hadamard 2x2 direta (FHAD 2x2) e a transformada Hadamard 2x2 inversa (IHAD 2x2).

As transformadas são o caminho crítico da predição intra-quadros, visto que para, que a predição intra-quadros possa realizar a predição de um bloco, é necessário ter reconstruído o bloco anterior do mesmo quadro (RICHARDSON, 2003). Entretanto, para ser reconstruído, o bloco deve passar pelos cálculos de transformadas diretas e inversas e quantização direta e inversa. Sendo assim, as arquiteturas desenvolvidas para os módulos de transformadas devem obter a melhor relação entre elevada taxa de processamento e baixa latência (tempo de resposta em ciclos) e, de preferência, com baixo consumo de recursos do hardware.

O objetivo deste trabalho é explorar diferentes modelos de arquiteturas multitransformadas, com elevada taxa de processamento, que possam realizar todos os cálculos definidos pelos módulos das transformadas diretas e inversas do padrão H.264/AVC.

A Seção 2 apresenta os materiais e métodos utilizados neste trabalho. A Seção 3 apresenta uma discussão sobre os resultados obtidos. Por fim, a Seção 4 conclui o trabalho.

2. MATERIAL E MÉTODOS

As equações dos cálculos realizados pelas transformadas FDCT 4x4, FHAD 4x4 e FHAD 2x2 são representadas, respectivamente, pelas equações (1), (2) e (3) (MALVAR, 2003).

$$Y = C_f W C_f^T \otimes E_f = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 2 & -1 \end{bmatrix} \right) \otimes \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix} \quad (1)$$

$$Y_D = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \right) / 2 \quad (2)$$

$$W_{QD} = \left(\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \right) \quad (3)$$

As definições matemáticas das transformadas inversas são muito semelhantes às apresentadas em (1), (2) e (3) e suas definições podem ser encontradas no padrão H.264/AVC (ITU-T, 2003). O cálculo da FHAD 2x2 e da IHAD 2x2 são idênticos (MALVAR, 2003) e, por isso, serão citadas apenas como HAD 2x2 neste artigo.

As equações acima foram decompostas em somas, subtrações e deslocamentos binários, e operações redundantes foram agrupadas em estágios de forma a obter um grande reaproveitamento de hardware.

A Tabela 1 mostra parte do algoritmo obtido para o cálculo das multitransformadas. Para tornar simples a implementação da arquitetura em forma de estágios de *pipeline*, o algoritmo foi dividido em quatro etapas (a, b, c, S).

Tabela 1. Algoritmo da Multitransformada.

4x4 FHAD	4x4 FDCT	4x4 IDCT	4x4 IHAD	2x2 HAD
$a_0 = X_0 + X_{12}$ $a_1 = X_4 + X_8$... $a_{15} = X_7 - X_{11}$	$a_0 = X_0 + X_{12}$ $a_1 = X_4 + X_8$... $a_{15} = X_7 - X_{11}$	$a_0 = X_0 + X_2$ $a_1 = X_0 - X_2$... $a_{15} = X_{13} + X_{15}/2$	$a_0 = X_0 + X_{12}$ $a_1 = X_4 + X_8$... $a_{15} = X_7 - X_{11}$	$a_0 = X_0 + X_2$ <i>don't care</i> ... <i>don't care</i>
$b_0 = a_0 + a_1$ $b_1 = a_2 + a_3$... $b_{15} = a_{14} - a_{15}$	$b_0 = a_0 + a_1$ $b_1 = a_2 + a_3$... $b_{15} = a_{14} - 2*a_{15}$	$b_0 = a_0 + a_3$ $b_1 = a_1 + a_2$... $b_{15} = a_{12} - a_{15}$	$b_0 = a_0 + a_1$ $b_1 = a_2 + a_3$... $b_{15} = a_{14} - a_{15}$	$b_0 = a_0 + 0$ <i>don't care</i> ... <i>don't care</i>
$c_0 = b_0 + b_3$ $c_1 = b_1 + b_2$... $c_{15} = b_{13} - b_{14}$	$c_0 = b_0 + b_3$ $c_1 = b_1 + b_2$... $c_{15} = b_{13} - b_{14}$	$c_0 = b_0 + b_8$ $c_1 = b_1 + b_9$... $c_{15} = b_7 + b_{15}/2$	$c_0 = b_0 + b_3$ $c_1 = b_1 + b_2$... $c_{15} = b_{13} - b_{14}$	$c_0 = b_0 + 0$ $c_1 = 0 + b_3$... <i>don't care</i>
$S_0 = (c_0 + c_1)/2$ $S_1 = (c_2 + c_3)/2$... $S_{15} = (c_{14} - c_{15})/2$	$S_0 = c_0 + c_1$ $S_1 = 2*c_2 + c_3$... $S_{15} = c_{14} - 2*c_{15}$	$S_0 = c_0 + c_{12}$ $S_1 = c_1 + c_{13}$... $S_{15} = c_3 - c_{15}$	$S_0 = c_0 + c_1$ $S_1 = c_2 + c_3$... $S_{15} = c_{14} - c_{15}$	$S_0 = c_0 + c_1$ <i>don't care</i> ... <i>don't care</i>

Por exemplo, o termo a_0 , que é o resultado final do primeiro bit no primeiro estágio, é obtido pela soma das entradas X_0 e X_{12} para FHAD 4x4, FDCT 4x4 e IHAD 4x4, enquanto que para IDCT 4x4 e HAD 2x2, é obtido através da soma de X_0 e X_2 . Os “*don't care*” encontrados no algoritmo da HAD 2x2 significam que parte do

hardware da multitransformada não é usado nos cálculos da HAD 2x2, devido à baixíssima complexidade desta transformada.

As arquiteturas desenvolvidas foram descritas em VHDL, sintetizadas para o dispositivo FPGA Altera Stratix II EP2S15F672C3 (ALTERA, 2009), com o auxílio da ferramenta Altera Quartus II (ALTERA, 2009) e validadas utilizando a ferramenta de simulação Mentor Graphics Modelsim 6.2g.

3. RESULTADOS E DISCUSSÃO

Diversas configurações de *pipeline* foram exploradas no decorrer desta investigação, dentre estas, serão apresentadas a versão sem *pipeline* (puramente combinacional) e a versão com quatro estágios de *pipeline* (*full pipelined*).

As arquiteturas desenvolvidas recebem amostras de 8 bits como entradas e geram saídas com 12 bits. Este crescimento na faixa dinâmica é necessário para prevenir o transbordo de representação que pode ocorrer nas operações de soma e subtração da arquitetura.

Na Tab. 2 estão apresentadas os resultados de latência, máxima frequência de operação, taxa de processamento e o número de quadros QHDTV processados por segundo para as duas arquiteturas. A Tab. 3 apresenta os resultados de consumo de recursos de hardware das arquiteturas, considerando número de ALUTs e registradores do FPGA.

Tabela 2. Resultados de latência e taxa de processamento.

Arquiteturas	Latência (ciclos)	Frequência (MHz)	Taxa de Processamento (M amostras/s)	Quadros QHDTV
1-estágio	1	98,61	1577,8	133,7
4-estágios	4	307,13	4914,1	416,6

Quadro QHDTV: 3840x2048 pixels

Tabela 3. Resultados de consumo de hardware.

Arquiteturas	Nº de ALUTs	Nº de Registradores Dedicados
1-estágio	1.769	320
4-estágios	1.394	800

Em relação ao desempenho, fica claro que a arquitetura de 4 estágios é superior, pois chega a ser 3 vezes mais rápida que a de 1 estágio (combinacional), o que ocorre devido ao menor período de relógio e ao caminho crítico reduzido que permitem uma maior frequência de operação. No entanto, esta arquitetura apresenta uma maior latência e um aumento no consumo de recursos hardware, ambos resultantes da utilização das barreiras de registradores requeridas pelo *pipeline*.

4. CONCLUSÕES

Este trabalho apresentou projetos arquiteturais de multitransformadas capazes de realizar os cálculos das transformadas direitas e inversas do padrão H.264/AVC de compressão de vídeo. O objetivo é reunir, em um único componente de hardware, todos os cálculos previstos pelo padrão, com o intuito de obter um alto desempenho e baixa latência, com um custo razoável em recursos de hardware.

Duas versões foram desenvolvidas e analisadas: (1) solução com apenas um estágio de *pipeline* e (2) solução com quatro estágios de *pipeline*. A solução com quatro estágios de *pipeline* possui o maior desempenho, mas também o maior consumo de recursos. Ainda assim, as duas soluções atingem uma taxa de processamento mínimo de 133 quadros QHDTV (3840x2048 pixels) por segundo.

Como trabalho futuro, pretende-se integrar o módulo das multitransformadas com o módulo das multiquantizações, ainda em desenvolvimento.

5. REFERÊNCIAS BIBLIOGRÁFICAS

JOINT VIDEO TEAM OF ITU-T AND ISO/IEC JTC 1. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 or ISO/IEC 14496-10 AVC), 2003.

RICHARDSON, I. **H.264/AVC and MPEG-4 Video Compression – Video Coding for Next-Generation Multimedia**. Chichester: John Wiley and Sons, 2003.

MALVAR, H., HALLAPURO, A., KARCZEWICZ, M., KEROFISKY, L. Low-Complexity Transform and Quantization in H.264/AVC. **IEEE Transactions on Circuits and Systems for Video Technology**, 2003, v.13, n.7, p.598-603.

ALTERA CORPORATION: Altera - The Programmable Solutions Company. Disponível em: <<http://www.altera.com>> Acesso em: 30 jun. 2009.