



## PROJETO DE HARDWARE PARA A QUANTIZAÇÃO DE ELEMENTOS DC COM FOCO NA PREDIÇÃO INTRA QUADROS DO PADRÃO DE COMPRESSÃO DE VÍDEO DO SISTEMA BRASILEIRO DE TV DIGITAL

**DORNELLES, Robson Sejanos Soares; SAMPAIO, Felipe Martin;**  
**PALOMINO, Daniel Munari; AGOSTINI, Luciano Volcan.**

*Grupo de Arquiteturas e Circuitos Integrados (GACI)  
Departamento de Informática – Universidade Federal de Pelotas  
Campus Universitário – Caixa Postal 354 – CEP 96010-900 - rdornelles.ifm@ufpel.edu.br*

### 1. INTRODUÇÃO

Este trabalho contextualiza-se na área de compressão de vídeo digital para o Sistema Brasileiro de TV Digital (SBTVD), e propõe soluções em hardware para o mais novo padrão de compressão de vídeo, o H.264/AVC (ITU-T, 2003).

Um vídeo é uma seqüência de imagens estáticas que se sucedem em grande velocidade, dando a impressão de movimento real. Tais imagens estáticas são chamadas de quadros (*frames*) e são formadas por um conjunto de pontos (*pixels*). Para realizar a compressão (codificação) de vídeo, é necessário codificar os quadros que formam o vídeo. Isto se faz possível dada a possibilidade de explorar as redundâncias de informação presentes nas imagens, reduzindo a quantidade de bits necessária para a representação desta informação. Em um mesmo quadro, por exemplo, existe o que é chamado de *redundância espacial*, que ocorre quando regiões do quadro são muito parecidas. Isso acontece, por exemplo, em uma cena que capture o céu azul. Nesta cena, talvez seja possível representar todo o céu com a informação obtida em apenas um pedaço do quadro.

No codificador do padrão H.264/AVC, a redundância espacial é explorada por um componente chamado de Predição Intra, que trabalha com blocos de 4x4, 8x8 e até 16x16 *pixels* (JVT, 2003). A Predição Intra codifica um bloco utilizando outros blocos semelhantes da imagem (blocos vizinhos) como referência. Entretanto, um bloco deve ter sido previamente codificado e reconstruído, ou seja, deve ter passado por um caminho de dados estabelecido pelo padrão H.264/AVC, para que possa servir como referência. Desse caminho de dados fazem parte os blocos das Transformadas Diretas (bloco T), da Quantização Direta (bloco Q), da Quantização Inversa (bloco  $Q^{-1}$ ) e das Transformadas Inversas (bloco  $T^{-1}$ ), que formam o *loop* (laço)  $TQQ^{-1}T^{-1}$ . A velocidade com que um bloco passa por esse *loop* tem relação estreita com o desempenho da Predição Intra: ela apenas volta a funcionar quando o bloco que acabou de ser codificado estiver reconstruído, de modo que possa ser

usado como referência. Isso só acontece depois que o bloco passa pelo *loop* TQQ<sup>-1</sup>T<sup>-1</sup> (RICHARDSON, 2003). Assim, torna-se interessante o projeto arquitetural de um *loop* com alta taxa de processamento e baixa latência, dedicado à Predição Intra.

O foco deste trabalho é o desenvolvimento de uma arquitetura para parte do módulo Q deste *loop* dedicado. A arquitetura implementada tem por objetivo realizar os cálculos relativos à quantização dos elementos DC, que são extraídos dos resultados do cálculo das transformadas Hadamard 2x2 e 4x4 diretas (do bloco T).

Este trabalho está dividido da seguinte maneira: a seção 2 apresenta o algoritmo da quantização, mostra o processo de desenvolvimento e a arquitetura projetada para a resolução do problema; a seção 3 apresenta os resultados da síntese; e a seção 4 conclui o trabalho.

## 2. MATERIAL E MÉTODOS

A quantização para os elementos DC é definida pelo padrão H.264/AVC por (1) (RICHARDSON, 2003). Esta é a equação principal da quantização.  $Y$  é o elemento a ser quantizado e  $Z$  é a saída da quantização. É importante observar que a quantização é realizada sobre o módulo do dado de entrada e, após o processo, o sinal da entrada é atribuído ao resultado calculado.

$$\begin{aligned} |Z_{D(i,j)}| &= (|Y_{D(i,j)}| \cdot MF_{(0,0)} + 2f) \gg (qbits + 1) \\ sign(Z_{D(i,j)}) &= sign(Y_{D(i,j)}) \end{aligned} \quad (1)$$

A especificação de (1) é função das equações (2), (3), (4) e (5).

$$MF(i, j) = \frac{PF(i, j)}{Qstep} \ll qbits \quad (2)$$

$$qbits = 15 + \lfloor QP/6 \rfloor \quad (3)$$

$$f = 2^{qbits} / 2 \quad (4)$$

$$PF = \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix} \quad \text{Onde } a = \frac{1}{2}, \text{ e } b = \sqrt{\frac{2}{5}} \quad (5)$$

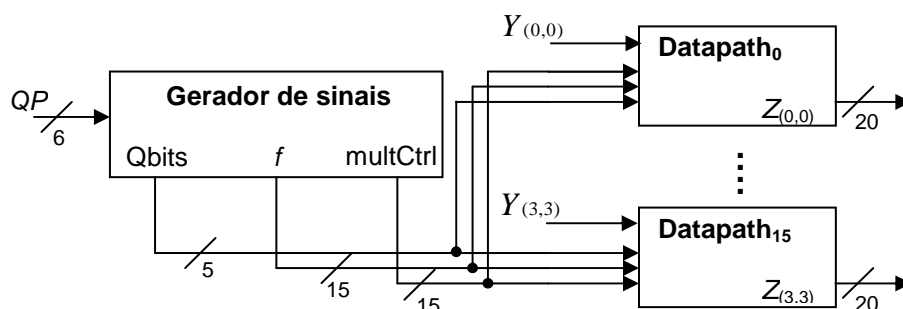
O parâmetro de quantização  $QP$  apresenta extrema importância no processo de quantização. O parâmetro de quantização apresenta valores entre 0 e 51 (RICHARDSON, 2003) e define o peso do processo de quantização. As variáveis  $qbits$  (3) e  $Qstep$  (Tabela 1) são funções direta de  $QP$ , enquanto  $MF$  (2) e  $f = 2^{qbits}/2$  são funções indiretas de  $QP$ . Desse modo, fazendo uso dos valores de  $QP$ , é possível calcular todos os valores existentes para todas variáveis internas ao processo de quantização. Com tais valores calculados, fica facilitado o desenvolvimento da solução arquitetural, pois é mais vantajoso armazenar os valores possíveis das variáveis em memória física e indexá-las com a variável  $QP$  do que calculá-las a cada novo cálculo da quantização.

**Tabela 1.** Relação entre o valor de Qstep e QP.

QP	0	1	2	3	4	5	6	...	12
Qstep	0,625	0,6875	0,8125	0,875	1	1,125	1,25	...	2,5

Em (2), é apresentado o cálculo da constante  $MF$ , que é utilizada em (1). É importante observar o comportamento da variável Qstep (Tabela 1) e da variável qbits (3). O valor de Qstep dobra a cada passo de seis no valor de QP, de modo que, por exemplo,  $Qstep_{(6)}$  é igual a  $Qstep_{(0)} \times 2$  (RICHARDSON, 2003). Por outro lado, cada vez que Qstep dobra, qbits é incrementada de uma unidade. Utilizando tais informações em (2), as modificações de qbits e Qstep (em função da variação de QP) acabam por se anular, ficando qbits sempre estável com o valor 15 e Qstep restrito as primeiras seis posições da tabela (Tabela 1) que define seus valores. Assim, para um dado PF (que é diretamente relacionado à posição original de uma amostra em seu bloco), existem apenas 18 valores possíveis para a variável  $MF$ . Visto que o hardware desenvolvido foca na quantização dos elementos DC, e sendo estes elementos todos referentes à posição (0,0), PF fica restrito à apenas um valor e  $MF$  é reduzido (para este projeto específico) a seis valores possíveis. Novamente, o armazenamento desses dados facilita e otimiza a criação do hardware final.

Assim, a arquitetura desenvolvida foi projetada em dois módulos principais: um **Gerador de Sinais** e dezesseis **Datapaths** (caminhos de dados), sendo capaz de realizar, em paralelo e em um único ciclo, a quantização de até 16 elementos DC provenientes das transformadas Hadamard 2x2 (utilizando apenas 4 *datapaths*) e da Hadamard 4x4 (utilizando os 16 *datapaths*).



**Figura 1.** Diagrama de Blocos da arquitetura desenvolvida.

É função do **Gerador de Sinais** prover os valores das variáveis envolvidas no processo de quantização. Os valores destas constantes não são calculados da maneira que são apresentados nas equações desta seção, visto que tais cálculos (como divisão e resto) são notadamente custosos quando implementados em hardware e acabariam por definir o gargalo da arquitetura. Tais valores serão previamente calculados, armazenados em memória e indexados pelo parâmetro  $QP$ .

Para fins de otimização, o processo de quantização não é realizado sobre o módulo do valor de entrada (eliminando duas operações de complemento de dois nas extremidades da arquitetura). Para tal, a variável  $f$  é armazenada e tratada diferentemente para valores positivos e negativos. A variável  $f$  é o ponto crítico para o tratamento do módulo da entrada, pois é adicionada ao resultado e, deste modo, o sinal da entrada interfere no resultado desta soma. Para contornar tal situação, é necessário somar 1 aos resultados provenientes de valores negativos.

Os **Datapaths** implementam a equação (1). O caminho crítico do *datapath* passa pela multiplicação inicial, que lida com entradas com larguras de bits elevadas

e, deste modo, acaba definindo um caminho crítico bastante longo. Em função de existirem apenas seis valores possíveis para a constante  $MF$ , é possível desenvolver um multiplicador especializado para tais constantes, decompondo a multiplicação em deslocamentos controlados, seguidos de uma árvore de somadores, sendo que o controle dos deslocadores fica sendo função de  $MF$  (e, logo, fica a cargo do **Gerador de Sinais**). Neste novo cenário e após a implementação, o caminho crítico do multiplicador especializado fica composto de apenas três somadores, o que representa um ganho considerável em comparação ao uso de um multiplicador genérico. Após a multiplicação, ainda existe uma soma (com a variável  $f$ ) e um deslocamento (controlado por  $qbits$ ). Tais componentes foram implementados de maneira usual, e a integração de todos estes componentes forma um **Datapath**, sendo 16 datapaths usados em toda a arquitetura.

### 3. RESULTADOS E DISCUSSÃO

A arquitetura apresentada na seção anterior foi descrita em VHDL e sintetizada para o FPGA Stratix II (modelo EP2S15F484C3), da Altera, usando a ferramenta Quartus II 8.0, também da Altera (ALTERA, 2008). Abaixo, os resultados:

**Tabela 2.** Resultados da síntese

Freqüência (MHz)	Taxa de Processamento (Bilhões de amostras / s)	Quadros QHDTV processados por segundo (3840x2048)	Consumo de Hardware ( <i>Look Up Tables</i> Adaptativas e <i>Dedicated Logic Registers</i> )
93.35	1,5	1012	5215 e 598

Da Tabela 2 é possível perceber que a arquitetura atinge uma taxa de processamento muito elevada, com latência de apenas um ciclo de relógio. Esta arquitetura atinge tempo real de processamento (30 quadros por segundo) para vídeos de resolução muito elevada, como QHDTV. O consumo de recursos de hardware representa apenas 11% da capacidade total do dispositivo FPGA utilizado. A elevada taxa de processamento associada à baixa latência são as características mais importantes para este projeto, que é focado no *loop* dedicado à Intra Predição.

### 4. CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou o projeto e os resultados de síntese de uma solução arquitetural paralela para a quantização dos elementos DC, focando em um *loop*  $TQQ^{-1}T^{-1}$  dedicado à Predição Intra do codificador do padrão H.264/AVC.

A partir dos resultados apresentados é possível concluir que a arquitetura proposta atinge os requisitos para utilização no *loop* dedicado para Predição Intra, pois apresenta uma taxa de processamento muito elevada e uma latência de apenas um ciclo de relógio. Por outro lado, a arquitetura desenvolvida também pode ser utilizada em um projeto de quantização convencional do codificador H.264/AVC que tenha o baixo consumo de energia como alvo, pois com uma freqüência de operação reduzida ainda é possível atingir tempo real.

Como trabalho futuro, será desenvolvido todo o *loop* dedicado à Predição Intra, utilizando processo de criação semelhante ao utilizado neste trabalho.

## 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.

ALTERA CORPORATION: Altera - The Programmable Solutions Company. Disponível em: <<http://www.altera.com>> Acesso em: 27 ago 2008.