

# Arquitetura de Alto Desempenho e Baixo Custo para o Módulo de Quantização Direta do Padrão H.264/AVC de Compressão de Vídeo Digital

Felipe Sampaio, Daniel Palomino, Robson Dornelles e Luciano Agostini

Universidade Federal de Pelotas - UFPel  
Grupo de Arquiteturas e Circuitos Integrados - GACI  
Pelotas, RS - Brasil  
{fsampaio.ifm, danielp.ifm, rdornelles.ifm, agostini}@ufpel.edu.br

**Abstract.** This work presents a dedicated hardware design of the Forward Quantization Module (Q module) of the H.264/AVC Video Coding Standard, using optimized multipliers. The goal of this design is to achieve high throughput rates combined with low hardware consumption. The architecture was described in VHDL and synthesized to the EP2S60F1020C3 Altera Stratix II FPGA and to the TSMC 0.18 $\mu$ m Standard Cell technology. The architecture reaches 364.2 MHz as maximum operation frequency. At this frequency, the architecture can process around 117 QHDTV frames (3840x2048 pixels) per second. The designed architecture can be used in low power and low cost applications, since it can process high resolution videos in real time even with very low operation frequencies and with low hardware consumption. In the comparison with related works, the designed Q module achieves the best results of throughput and hardware consumption.

**Palavras-chave:** Padrão H.264/AVC, quantização, compressão de vídeo, projeto VLSI.

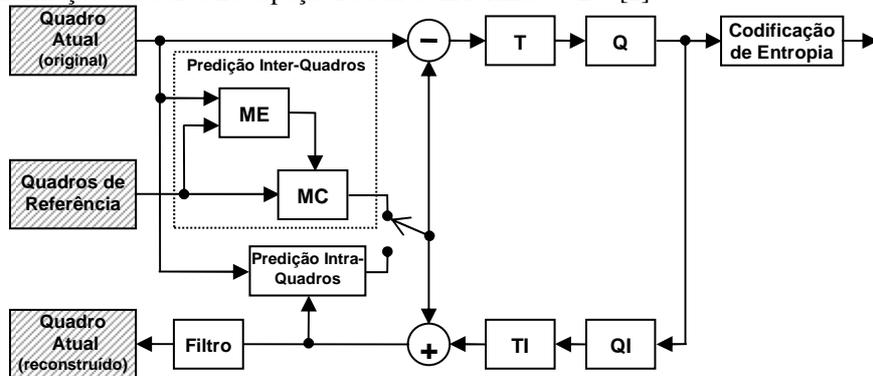
## 1 Introdução

O H.264/AVC é o mais novo padrão de compressão de vídeo [1]. Ele foi desenvolvido por especialistas da ISO/IEC e do ITU-T e tem por objetivo dobrar as taxas de compressão alcançadas pelos padrões anteriores.

A compressão de vídeo é extremamente importante dada a grande quantidade de informação presente nos vídeos de alta definição. O padrão H.264/AVC apresenta ferramentas eficientes para a compressão de vídeo, entretanto, codificadores H.264/AVC possuem elevada complexidade computacional. Desse modo, soluções em hardware se fazem necessárias para que seja possível codificar vídeos de alta resolução em tempo real (30 quadros por segundo).

No processo de compressão, o padrão explora os diferentes tipos de redundância encontrados em um vídeo: espacial, temporal e entrópica. O diagrama de blocos do codificador pode ser visto na Fig. 1. O padrão oferece diversos perfis de codificação,

sendo que este trabalho tem como escopo o perfil *main*, que representa as informações de cores no espaço YCbCr com formato 4:2:0 [2].



**Figura 1.** Diagrama em blocos do codificador H.264/AVC

A Predição Intra-Quadros do padrão H.264/AVC contribui para reduzir a redundância espacial de um vídeo. A Predição Intra-Quadros é uma inovação do padrão, cuja função é explorar a similaridade entre blocos de pixels dentro de um quadro, utilizando a informação das bordas dos blocos vizinhos já codificados para representar os blocos adjacentes. Desta maneira, o bloco predito é criado através de uma cópia destas bordas. O padrão define uma grande variedade de modos de cópia. Considerando a camada de luminância (Y): são nove modos de cópia para blocos 4x4 e quatro modos para os blocos de tamanho 16x16. Para as camadas de croma (Cb e Cr) existem apenas quatro modos, visto que sempre são usados blocos 8x8 (semelhantes aos modos de luminância 16x16) [2].

A redundância temporal é explorada pela Predição Inter-Quadros, composta por dois módulos: a Estimativa de Movimento (ME) e a Compensação de Movimento (MC). Neste caso, a codificação do bloco atual é feita a partir de informações dos quadros previamente codificados. A ME faz uma busca nos quadros de referências pelo bloco que mais se assemelha ao bloco atual e gera uma informação de movimento (vetor de movimento), indicando qual bloco do quadro de referência que será utilizado, enquanto que a MC remonta o quadro atual a partir dos vetores de movimento gerados pela ME [2].

Quando as etapas de predição geram um novo bloco, a diferença entre o bloco predito e o bloco original gera uma informação residual que deve ser transformada e quantizada a fim de aumentar as taxas de compressão. Entretanto, a quantização é um processo que gera perdas, o que significa que o bloco reconstruído no decodificador será diferente do bloco predito pelo codificador. Como este bloco será utilizado como referência pelo decodificador no processo de decodificação, ele também deve ser utilizado como referência pelo codificador, para evitar inconsistências no processo de codificação. Assim, é necessário introduzir os módulos de Quantização Inversa (QI) e de Transformadas Inversas (TI) também no codificador, como pode ser observado na Fig. 1.

Este trabalho propõe uma arquitetura de alto desempenho e de baixo custo para o módulo de Quantização do codificador H.264/AVC. A arquitetura é capaz de

processar até quatro amostras por ciclo (uma linha de um bloco 4x4) e atingir um desempenho que garanta o processamento de vídeos de alta resolução em tempo real mesmo em baixas frequências de operação. A arquitetura foi descrita em VHDL e sintetizada para um dispositivo FPGA Stratix II da Altera e também para a tecnologia *Standard Cell* TSMC 0.18 $\mu$ m.

Este trabalho é organizado como se segue: a Seção 2 descreve o módulo Q e apresenta sua definição; a Seção 3 apresenta a arquitetura desenvolvida; a Seção 4 apresenta os resultados de síntese da arquitetura para as tecnologias já citadas; a Seção 5 traz uma comparação com trabalhos relacionados da literatura; e, finalmente, a Seção 6 conclui este trabalho e apresenta os trabalhos futuros a serem desenvolvidos.

## 2 Quantização no Padrão H.264/AVC

O padrão H.264/AVC define que a informação residual oriunda dos módulos de predição seja transformada do domínio espacial para os domínios das frequências, para que sejam mais bem manipuladas pela etapa de quantização. A quantização faz um corte nas frequências para as quais o sistema visual humano é menos sensível, gerando valores que facilitam o processo de codificação de entropia [2], gerando altas taxas de compressão.

Os módulos de quantização direta recebem os coeficientes oriundos das transformadas. Após serem transformadas, as amostras de um bloco 4x4 são chamadas de coeficientes: o elemento (0,0) do bloco é o coeficiente DC e os demais elementos são chamados de coeficientes AC [2].

O padrão H.264/AVC é o primeiro padrão de compressão de vídeo que utiliza três diferentes transformadas nos módulos T e TI da Fig. 1. No módulo T, todos os coeficientes de entrada são processados pela DCT direta (FDCT 4x4). Adicionalmente, alguns dos coeficientes DC são processados por uma transformada Hadamard, para explorar uma correlação residual presente entre os elementos DC de diferentes blocos 4x4. Todos os coeficientes DC de blocos de crominância são processados pela Hadamard 2x2 (HAD 2x2), enquanto que os coeficientes DC de blocos de luminância que foram gerados pelo modo Intra\_6x16 são processados pela transformada Hadamard 4x4 FHAD 4x4 [2].

O processo de quantização é realizado de diferentes maneiras, de acordo com o componente de cor (Y, Cb ou Cr) e com o modo de predição escolhido.

Quando o modo de predição escolhido for Intra\_4x4, ambos os coeficientes AC e DC do bloco são quantizados segundo a Equação (1) [3]. Esse cálculo também é aplicado para todos os coeficientes AC para os modos Intra\_16x16 e Croma\_8x8. Todo o resíduo gerado pela Predição Inter-Quadros também é quantizado desta maneira.

$$\begin{aligned} |Z_{(i,j)}| &= (|W_{(i,j)}| \cdot MF + f) \gg qbits \\ \text{sign}(Z_{(i,j)}) &= \text{sign}(W_{(i,j)}) \end{aligned} \quad (1)$$

Para todos os coeficientes DC de luminância cujo modo de predição for Intra\_16x16, e para todos os coeficientes DC de crominância, o cálculo a ser realizado pela quantização está representado pela Equação (2) [3]. Deste modo, todos os coeficientes de saída das transformadas Hadamard são processados dessa forma.

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

As constantes utilizadas pelas equações (1) e (2) estão definidas na Tab. 1 e nas equações (3) e (4). Todos esses valores são função do parâmetro de quantização QP (*Quantization Parameter*). O QP define a “força” do processo de quantização, de modo que um alto valor para este parâmetro resulta em um processo de quantização com mais perdas na qualidade da imagem, o que, no entanto, gera maiores taxas de compressão. Um baixo valor para o parâmetro de quantização ocasiona uma maior qualidade para o vídeo codificado, contudo, menores taxas de compressão são alcançadas. O QP é gerado pelo controle global do codificador, serve como entrada para o módulo Q e seu valor pode variar de 0 até 51. O operador % na Tab. 1 é o operador de resto, ou seja, a operação  $a \% b$  resulta no resto da divisão de  $a$  por  $b$ .

**Tabela 1.** Valores de MF de acordo com o QP

QP%6	(PF0) Posições (0,0),(2,0),(2,2),(0,2)	(PF1) Posições (1,1),(1,3),(3,1),(3,3)	(PF2) Outras Posições
0	13107	5243	8066
1	11916	4660	7490
2	10082	4194	6554
3	9362	3647	5825
4	8192	3355	5243
5	7282	2893	4559

$$f = 2^{qbits}/3, \text{ para a Predição Intra-Quadros} \quad (3)$$

$$f = 2^{qbits}/6, \text{ para a Predição Inter-Quadros}$$

$$qbits = 15 + \lfloor QP/6 \rfloor \quad (4)$$

**Exemplo:** Uma situação hipotética onde o valor de entrada tem o valor  $Y=150$ , na posição (1,1) do bloco 4x4 e que tenha sido predito com o modo Intra\_4x4. Considerando um  $QP=16$ , a operação  $QP \% 6$  resultaria em 4, o que significa que o valor de MF, segundo a Tab. 1, seria 3355. Pelas equações (3) e (4) é possível concluir que  $qbits=17$  e  $f=43690$ . Deste modo, aplicando estes valores na Equação (1) resultará no valor quantizado  $Z=4$ .

As definições mais detalhadas dos cálculos utilizados nos processos de quantização, bem como a descrição original de como as constantes utilizadas são obtidas, podem ser encontradas no padrão H.264/AVC [1]. O processo de derivação das fórmulas originais pode ser encontrado em [2].

### 3 Arquitetura Desenvolvida

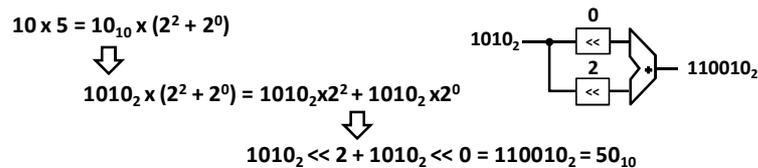
Como visto na seção anterior, o processo de quantização direta definido no padrão H.264/AVC é, basicamente, composto pela multiplicação do valor de entrada por uma constante (MF), a soma do resultado com outra constante ( $f$  ou  $2f$ ) e o deslocamento controlado por outra constante ( $qbits$  ou  $qbits+1$ ). Deste modo, o caminho de dados de uma arquitetura genérica para realizar a quantização direta deve conter: um multiplicador, um somador e um deslocador controlado. Dentre as operações realizadas, a multiplicação é aquela que possui a maior complexidade e, portanto, as otimizações feitas nesses multiplicadores são de extrema importância para os resultados do módulo inteiro.

O valor da constante MF depende da posição do coeficiente alvo no bloco 4x4. Com base na Tab. 1, nota-se que existem três conjuntos de posições onde o MF se mantém o mesmo. Deste modo, é possível projetar multiplicadores dedicados para cada conjunto de valores de MF, de modo a reduzir o impacto da multiplicação no caminho crítico da arquitetura. Assim, três diferentes caminhos de dados foram desenvolvidos, diferindo apenas no multiplicador dedicado utilizado de acordo com a posição do coeficiente alvo da quantização.

Devido às poucas possibilidades de valores que as constantes podem assumir, tais valores foram previamente calculados e armazenados em memória. A unidade de controle usa o valor de QP para realizar o endereçamento da memória.

As soluções arquiteturais para os multiplicadores foram sendo analisadas gradativamente, partindo de (1) um multiplicador matricial convencional, passando por (2) um multiplicador utilizando a decomposição em somas e deslocamentos e, por fim, aproveitando-se das demais operações que compõem o processo de quantização, chega-se a um (3) multiplicador dedicado que incorpora a soma com a constante  $f$  e o deslocamento de 15 bits.

A decomposição da operação de multiplicação em operações de somas e deslocamentos se vale da propriedade aritmética da distributividade entre soma e multiplicação, como é mostrado na Fig. 2.

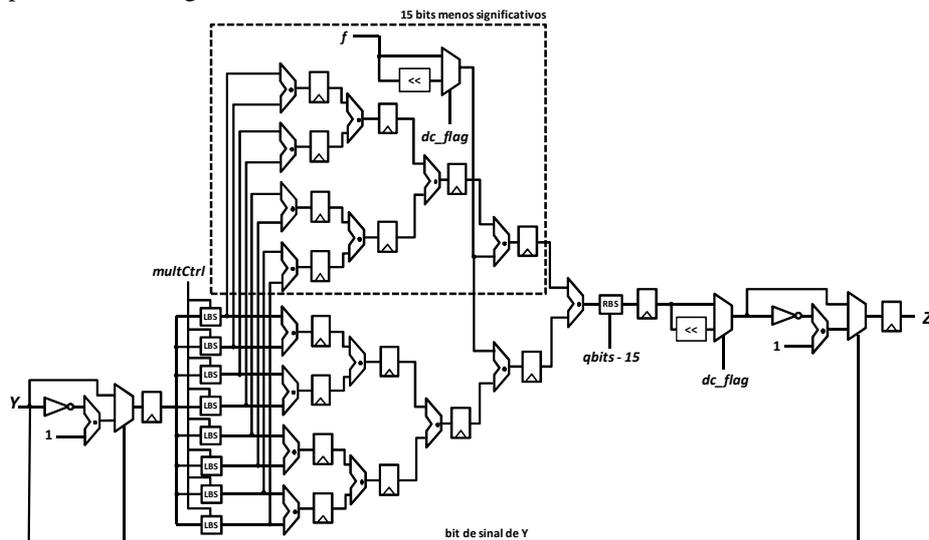


**Figura 2.** Exemplo de utilização da propriedade distributiva para decompor uma operação de multiplicação em operações de somas e deslocamentos.

O cálculo da quantização define um deslocamento à direita de no mínimo 15 bits, ou seja, após a soma com a constante  $f$ , os 15 bits menos significativos devem ser descartados. Na solução desenvolvida neste trabalho, esse deslocamento foi aplicado diretamente aos valores de entrada da arquitetura, antes mesmo da multiplicação. Deste modo, a largura dos somadores foi reduzida, ocasionando em uma redução no caminho crítico total. Para evitar erros, uma verificação em paralelo é feita com os 15

bits descartados. Se essa verificação acusar que houve *overflow*, então ele será adicionado ao resultado final.

A arquitetura desenvolvida opera sobre entradas de 15 bits, e sua descrição RTL é apresentada na Fig. 3.



**Figura 3.** Caminho de dados do módulo Q desenvolvido neste trabalho.

Os somadores das extremidades mostrados na Fig. 3 são necessários para a operação de complemento de dois aplicada na entrada e na saída da arquitetura, visto que as operações de quantização são aplicadas sobre o módulo do valor de entrada.

Os sinais  $multCtrl$ ,  $dc\_flag$  e  $qbits$  são entregues ao caminho de dados pela unidade de controle, e seus valores são definidos em função de qual modo de predição foi utilizado na codificação do bloco, qual coeficiente está sendo processado (DC ou AC), qual o valor de QP e qual é a posição da amostra no bloco 4x4.

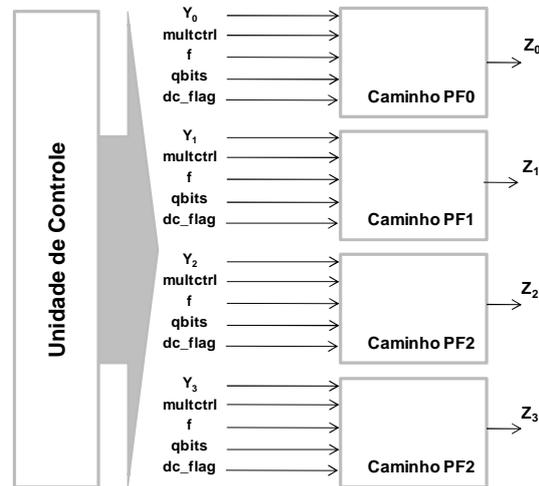
Como as operações de deslocamentos geram bits zeros na parte menos significativa dos dados, os somadores que compõe a árvore de operadores puderam ser simplificados. Isso não acarreta em um ganho em termos de caminho crítico, visto que o último operador terá que operar sobre todos os bits. Entretanto, há uma significativa redução no consumo de hardware, visto que, considerando uma implementação *Ripple Carry* para os somadores, o número de somadores completos de 1 bit é substancialmente reduzido com essas simplificações.

Considerando o caminho de dados que realiza a quantização para a posição (0,0) e a largura de 15 bits na entrada, essas simplificações correspondem a uma redução de cerca de 22% no número de somadores completos de 1 bit.

Cada linha de somadores da arquitetura foi isolada entre barreiras temporais (registradores), a fim de se obter o máximo de desempenho possível. Deste modo, foi projetado um *pipeline* de sete estágios e o caminho crítico é representado pelo atraso do somador com a maior largura de bits.

A arquitetura completa é composta por quatro caminhos de dados em paralelo, conforme apresentado na Fig. 4. Deste modo, a arquitetura consome quatro

coeficientes gerados pelo módulo T a cada ciclo de relógio. Os nomes dos caminhos de dados na Fig. 4 (PF0, PF1 e PF2) são relacionados com cada tipo de multiplicação suportada e estão associados aos nomes apresentados na Tab. 1.



**Figura 4.** Arquitetura completa para o módulo Q.

É importante notar que um caminho de dados PF2 é utilizado duas vezes na arquitetura completa. Isto ocorre porque este caminho é responsável por processar oito dos 16 coeficientes do bloco 4x4, como mostrado na Tab. 1. Os caminhos PF0 e PF1, também conforme a Tab. 1, são responsáveis por processar quatro diferentes coeficientes cada um. Assim, com a duplicação de PF2 e com um caminho dedicado para PF0 e outro para PF1, cada caminho de dados fica responsável por realizar o cálculo de quatro coeficientes específicos.

A latência da arquitetura é de sete ciclos de relógio, ou seja, haverá uma saída válida sete ciclos após haver um dado válido na entrada da arquitetura. Após esses ciclos de latência, para cada novo ciclo de relógio são entregues quatro coeficientes quantizados na saída.

Para o bloco 2x2 de coeficientes DC de crominância que é gerado pela HAD 2x2, a arquitetura leva quatro ciclos de relógio para consumi-lo completamente, visto que todos os coeficientes devem ser aplicados ao mesmo caminho de dados (PF0). Da mesma maneira, o bloco 4x4 de coeficientes DC oriundos da transformada FHAD 4x4 leva 16 ciclos de relógio para ser consumido pela arquitetura, também pelo caminho PF0. Para os dados de saída da FDCT 4x4, todos os caminhos são utilizados e em quatro ciclos de relógio todos os coeficientes são entregues ao módulo Q.

## 4 Resultados de Síntese

A arquitetura desenvolvida foi descrita em VHDL e sintetizada para o dispositivo FPGA EP2S60F1020C3 da família Stratix II da Altera [4] e para a biblioteca de células lógicas TSMC 0,18 $\mu$ m *Standard Cells* [5]. A Tab. 2 mostra os resultados de

síntese obtidos em termos de máxima frequência de operação e consumo de hardware, para ambas as tecnologias de utilizadas.

**Tabela 2.** Resultados de síntese para as duas tecnologias alvos.

<b>Módulo Q Completo</b>		
<b>Stratix II FPGA</b>	<b>Freq. (MHz)</b>	291,7
	<b>#ALUTs</b>	1,308 (3%)
	<b>#DLRs</b>	956 (2%)
<b>TSMC 0,18µm</b>	<b>Freq. (MHz)</b>	346,2
	<b>#Gate</b>	14,5k

Como já esperado, os resultados de síntese para *standard cells* se mostraram os melhores. Nesta tecnologia, a arquitetura atingiu 346,2 MHz como frequência máxima de operação, superando em 19% a frequência atingida com o dispositivo FPGA.

A arquitetura, quando mapeada para FPGA, utiliza 1.308 ALUTs e 956 registradores dedicados do dispositivo. Estes dados representam apenas 3% dos recursos disponíveis no dispositivo FPGA alvo.

Com estes resultados, foi possível a estimativa de processamento da arquitetura desenvolvida, considerando vídeos de alta resolução. A Tab. 3 mostra esses dados de desempenho para a síntese em *standard cells*.

**Tabela 3.** Taxas de processamento para a implementação em *standard cells*.

	<b>Taxa de Processamento (bilhões coef./s)</b>	<b>Quadros QHDTV/s</b>	<b>Quadros HDTV/s</b>	<b>Freq. Mín HDTV a 30qps (MHz)</b>
<b>Módulo Q</b>	1,3848	117,4	445,2	23,3

Os resultados obtidos mostram que a solução desenvolvida atinge facilmente processamento em tempo real para vídeos de altíssima resolução, como QHDTV (3840x2048 pixels), quando opera na frequência máxima de operação.

A arquitetura proposta também é recomendada quando a restrição imposta está relacionada ao consumo de energia, visto que ela é capaz de processar vídeos de alta resolução mesmo com baixas frequências. Como mostrado na Tab. 3, é suficiente uma frequência de 23,3 MHz para que a arquitetura processe vídeos de resolução HDTV (1920x1080 pixels) em tempo real (30 quadros por segundo).

Um gargalo que limita o desempenho dos codificadores H.264/AVC está presente módulo de Predição Intra-Quadros. Este módulo codifica um quadro internamente, utilizando informações previamente codificadas como referência para representar outras áreas do quadro. Isso gera uma grave dependência de dados, visto que essas referências devem passar pela etapa de reconstrução (módulos T, Q, QI e TI) para, então, estarem disponíveis. Neste meio tempo, o Preditor Intra-Quadros fica inoperante. Deste modo, arquiteturas projetadas para este caminho de reconstrução devem possuir elevadas taxas de processamento e utilizar o menor número de ciclos possível para processar as informações necessárias. O módulo de quantização

desenvolvido neste trabalho é uma boa alternativa para este gargalo, já que ele apresenta uma taxa de processamento muito elevada.

## 5 Comparações com Trabalhos Relacionados

A comparação com trabalhos relacionados da literatura não é uma tarefa trivial. Isso ocorre devido ao objetivo particular de cada trabalho de atingir melhores resultados considerando um conjunto de restrições específicas.

A comparação aqui apresentada foi realizada considerando dois trabalhos publicados na literatura: [6] e [7]. Kordasiewicz [6] apresenta duas soluções arquiteturais para o módulo Q: uma com otimizações que buscam diminuir o consumo de hardware e outra com otimizações que visam altas frequências de operação. Já o trabalho proposto por Zhang [7] apresenta uma solução também com multiplicadores dedicados, mas não faz o projeto de um módulo Q completo.

O trabalho [7] apresenta seis diferentes multiplicadores para cada conjunto de posições do bloco 4x4, enquanto que este trabalho propõe somente um multiplicador capaz de realizar as mesmas operações, o que reduz drasticamente o consumo de hardware do módulo Q desenvolvido, quando comparado a um módulo que utilize os multiplicadores propostos por [7]. Comparações mais consistentes com o trabalho de Zhang não foram realizadas por que não foram apresentados dados de síntese para o módulo Q completo.

A Tab. 4 mostra a comparação da solução proposta neste trabalho com as duas soluções arquiteturais propostas por [6].

**Tabela 4.** Comparação com trabalhos relacionados.

	<b>Este Trabalho</b>	<b>Kordasiewicz [6] área</b>	<b>Kordasiewicz [6] atraso crítico</b>
<b>Tecnologia</b>	<b>TSMC 0,18<math>\mu</math>m</b>	TSMC 0,18 $\mu$ m	TSMC 0,18 $\mu$ m
<b>Nível de Paralelismo</b>	<b>4</b>	1	16
<b>Taxa de Processamento (bilhões de coef./s)</b>	<b>1,385</b>	0,02	1,085
<b>Área (milhares de Gates)</b>	<b>14,5</b>	1,75	39,9
<b>Processamento/Área</b>	<b>95,5</b>	11,4	27,2

Com relação à solução [6] com otimizações em área, os dados de consumo de hardware da arquitetura desenvolvida neste trabalho são maiores. Isto é justificado pelo fato da arquitetura [6] ter sido projetada com paralelismo de um coeficiente por ciclo. Deste modo, apenas um caminho de dados é utilizado. Esse baixo nível de paralelismo implica em uma baixa taxa de processamento, cerca de 70 vezes menor do que a taxa atingida pela arquitetura desenvolvida neste trabalho.

Considerando a solução de [6] com otimizações para reduzir o atraso crítico, a taxa de processamento se aproxima dos resultados obtidos neste trabalho, já que foi utilizado um maior nível de paralelismo (16 coeficientes por ciclo). Esse aumento no paralelismo elevou o consumo de hardware em 23 vezes com relação à solução com otimizações em área. Mesmo com este aumento no paralelismo, a taxa de processamento atingida é ainda cerca de 1,2 vezes menor do que a deste trabalho.

Apesar de haver um ganho em hardware na solução otimizada para área proposta por [6], a arquitetura desenvolvida neste trabalho atinge os melhores resultados na relação entre taxa de processamento e área consumida com relação às duas soluções propostas por [6].

## 6 Conclusões e Trabalhos Futuros

Este artigo apresentou uma arquitetura de alto desempenho e baixo custo para o módulo de quantização direta do padrão H.264/AVC de compressão de vídeo. Foi utilizado um paralelismo de quatro amostras por ciclo e explorado ao máximo o uso de *pipeline* para acelerar o processamento dos coeficientes oriundos do módulo das transformadas. A arquitetura foi descrita em VHDL e sintetizada para duas diferentes tecnologias: FPGA Stratix II da Altera e TSMC 0,18 $\mu$ m. Os resultados de síntese apontaram uma frequência máxima de operação de 346,2 MHz. Deste modo, a arquitetura é capaz de processar cerca de 117,4 quadros QHDTV por segundo. Para aplicações com restrições de consumo de energia, a solução proposta neste trabalho atinge processamento de tempo real mesmo em baixas frequências.

Como trabalho futuro, pretende-se a adaptação desta arquitetura para que ela possa ser utilizada na solução do problema de dependência de dados imposto pela Predição Intra-Quadros.

## Referências

1. ITU-T: Recommendation H.264/AVC (11/2007). Advanced Video Coding for Generic Audiovisual Services. (2007)
2. Richardson, I.: H.264/AVC and MPEG-4 Video Compression – Video Coding for Next-Generation Multimedia. John Wiley&Sons, Chichester. (2003)
3. Malvar, H.S.; Hallapuro, A.; Karczewicz, M.; Kerofsky, L.: Low-complexity transform and quantization in H.264/AVC. In: IEEE Transactions on Circuits and Systems for Video Technology. vol 13. (2003)
4. Altera Corporation: Altera: The Programmable Solutions Company. Available at: [www.altera.com](http://www.altera.com).
5. Artisan Components: TSMC 0.18  $\mu$ m 1.8-Volt SAGE-XTM Standard Cell Library Databook. (2001)
6. Kordasiewicz, R.C., Shirani, S.: Asic and FPGA Implementations of H.264 DCT and Quantization Blocks. In: IEEE Int. Conf. on Image Processing, IEEE, Piscataway, (2005)
7. Zhang, Y., Jiang, G., Yi, W., Yu, M., Li, F., Jiang, Z. and Liu, W.: An Improved Design of Quantization for H.264 Video Coding Standard. In: 8th International Conference on Signal Processing, IEEE, Piscataway, vol. 2. (2006)