ARQUITETURA COMPLETAMENTE PARALELA PARA O BLOCO DAS TRANSFORMADAS DIRETAS DO PADRÃO H.264/AVC DE COMPRESSÃO DE VÍDEO

Roger Porto[1], *Marcelo Porto*[1], *Fernanda Kastensmidt*[1], *Marcelo Lubaszewski*[1], *Luciano Agostini*[1,2], *Sergio Bampi*[1]

[1] Grupo de Microeletrônica – II – UFRGS – Porto Alegre, Brasil
[1, 2] Grupo de Arquiteturas e Circuitos Integrados – DInfo – UFPel – Pelotas, Brasil

{recporto, msporto, fglima, luba, agostini, bampi}@inf.ufrgs.br

RESUMO

Este trabalho apresenta uma implementação em hardware totalmente paralela para o bloco das transformadas diretas do padrão de compressão de vídeo H.264/AVC. As três transformadas que compõem o bloco foram implementadas, assim como a estrutura de sincronização dos dados de entrada e saída do bloco. A arquitetura do bloco T desenvolvida neste trabalho, quando mapeada em um FPGA Xilinx, é capaz de processar até 1561 quadros HDTV por segundo, podendo ser utilizada em um codificador H.264/AVC para HDTV em tempo real.

1. INTRODUÇÃO

A compressão de vídeos digitais é um assunto bastante explorado na atualidade, tanto pela academia, quanto pela indústria. Este interesse ocorre porque a compressão e descompressão deste tipo de mídia estão presentes em diversos equipamentos atuais, como: celulares, TVs digitais, DVD players, câmeras digitais, entre outros.

O padrão de compressão de vídeo H.264/AVC [1] é o mais novo padrão de compressão de vídeo e foi desenvolvido com o objetivo de dobrar a taxa de compressão em relação aos demais padrões existentes até então. O padrão H.264/AVC foi desenvolvido pelo JVT, que foi formado a partir de uma união entre os especialistas do VCEG da ITU-T e do MPEG da ISO/IEC [2] [3]. O padrão H.264/AVC atingiu seu objetivo, mas, para tanto, foi necessário um grande aumento na complexidade computacional de suas operações. Este aumento de complexidade impede, pelo menos na tecnologia atual, a utilização de codecs H.264/AVC implementados em software quando as resoluções são elevadas ou quando se deseja tempo real.

Este trabalho apresenta uma implementação em hardware para um dos blocos que compõem o codificador do padrão H.264/AVC, o bloco das transformadas diretas (também chamado de bloco T).

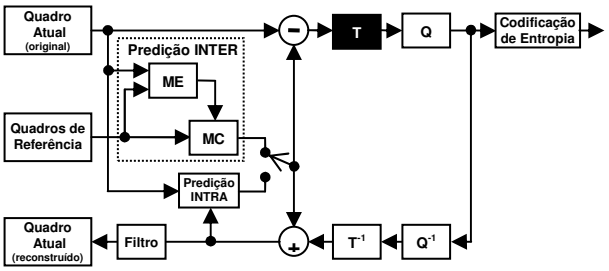

Figura 1 – Diagrama em blocos do codificador H.264/AVC

2. O BLOCO DAS TRANSFORMADAS DIRETAS DO PADRÃO H.264/AVC

O diagrama em blocos de um codificador H.264/AVC é apresentado na fig. 1. Os principais blocos do codificador H.264/AVC são: estimação de movimento (ME), compensação de movimento (MC), predição intra, transformadas diretas (T) e inversas (T^{-1}), quantização direta (Q) e inversa (Q^{-1}), filtro e codificação de entropia [2].

Como o objetivo deste trabalho foi desenvolver uma arquitetura totalmente paralela pata o bloco T do codificador H.264/AVC (destacado na fig. 1), os demais blocos não serão detalhados.

O bloco T é responsável pelas transformadas diretas e está presente apenas na parte de codificação do padrão H.264/AVC. Uma transformada discreta do cosseno em duas dimensões e duas transformadas Hadamard são as transformadas diretas que compõem este bloco. Todas as transformadas utilizadas pelo padrão H.264/AVC são inteiras [4].

As entradas para o bloco T são blocos 4x4 de resíduos gerados pela etapa de predição intra ou inter quadro. A esses dados é aplicada a DCT 2-D direta (FDCT 4x4). O cálculo da FDCT 4x4 inteira é definido no padrão H.264/AVC como está apresentado em (1) [4].

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

Em (1), **X** é a matriz 4x4 de entrada, $\mathbf{C_f}$ é a matriz da FDCT inteira em uma dimensão, $\mathbf{C_{fT}}$ é a transposta da matriz da FDCT e $\mathbf{E_f}$ é a matriz de fatores de escala. O símbolo ⊗ na equação indica uma multiplicação escalar.

O cálculo da FDCT 4X4 transfere para o bloco de quantização (Q) a tarefa de realizar a multiplicação escalar por $\mathbf{E_f}$. A quantização é o passo que segue a aplicação das transformadas diretas na compressão H.264/AVC. Esta tarefa adicional não implica em aumento na complexidade do bloco Q [2].

O cálculo da FDCT 4x4 é aplicado sobre todos os dados de entrada. No caso de amostras com informação de crominância ou com informação de luminância cuja predição tenha sido do tipo INTRA 16x16, um cálculo adicional é realizado. Neste caso, é aplicada uma transformada Hadamard 4x4 sob os coeficientes DC dos blocos de luminância, enquanto que aos blocos de crominância é aplicada uma transformada Hadamard 2x2. A transformada Hadamard explora uma correlação residual que ainda permanece sobre os coeficientes resultantes da FDCT 4x4.

O cálculo da Hadamard 4x4 direta definida pelo padrão H.264/AVC está apresentado em (2) [4].

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

O cálculo da transformada Hadamard 2x2 definido pelo padrão H.264/AVC está apresentado em (3) [4].

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

O bloco T, sendo formado por estas três transformadas, deve sincronizar a operação entre elas, de modo a gerar o fluxo correto de dados na sua saída. A ordem de processamento está apresentada na fig. 2. Inicialmente, o macrobloco de luminância (Y) passa pela FDCT 4x4. Se o modo é intra 16x16, os elementos DC das matrizes 4x4 resultantes da FDCT 4x4 (bloco -1 na fig.2) passam pela FHAD 4x4. Neste caso, primeiramente é enviado para saída o bloco –1, com os resultados da FHAD 4x4. Depois, os elementos AC são enviados para a saída (blocos 0 a 15 na fig. 2). Se o modo não for intra 16x16, os blocos 0 a 15 são enviados diretamente para a saída e a Hadamard 4x4 não é aplicada. Os blocos 16 e 17 são compostos de resultados da aplicação da FHAD 2x2 sobre os elementos DC de crominância e são enviados nesta ordem para a saída. Finalmente, os coeficientes AC de crominância são enviados para a saída (blocos 18 a 25 na fig. 2) [2].

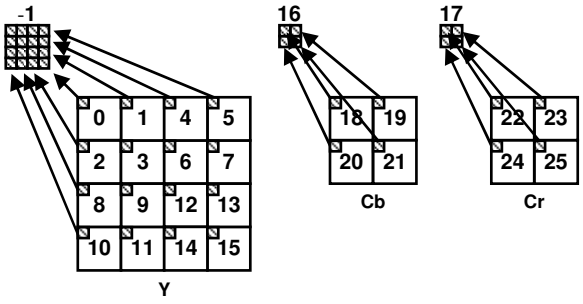

Figura 2 – Ordem de processamento de amostras pelo bloco T

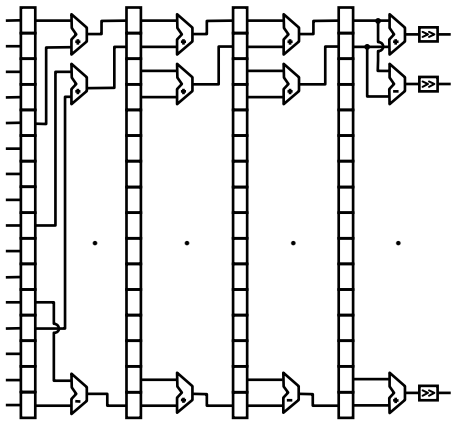

Figura 3 – Arquitetura da Hadamard 4x4 direta

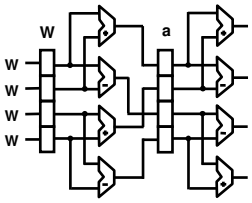

Figura 4 – Arquitetura da Hadamard 2x2 direta

3. ARQUITETURA DAS TRANSFORMADAS

As transformadas apresentadas neste trabalho foram desenvolvidas em arquiteturas totalmente paralelas. Dessa forma, a FDCT 4x4 e a FHAD 4x4 consomem 16 amostras a cada ciclo, enquanto que a FHAD 2x2 consome 4 amostras por ciclo.

A partir das características do algoritmo desenvolvido com base nas definições (1), (2) e (3), as arquiteturas totalmente paralelas para as transformadas FDCT 4x4, FHAD 4x4 e FHAD 2x2, foram desenvolvidas.

As arquiteturas 4x4 foram implementadas com 4 estágios de pipeline, onde cada estágio possui 16 operadores. A fig. 3 ilustra a arquitetura da FHAD 4x4. Como a arquitetura da FDCT 4x4 é similar à da FHAD 4x4, ela não será apresentada neste artigo.

A arquitetura da FHAD 2x2 foi desenvolvida em um pipeline de dois estágios e está apresentada na fig. 4.

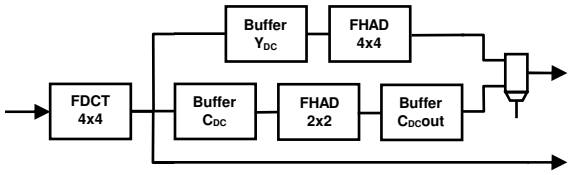

Figura 5 – Diagrama em blocos do bloco T paralelo

4. ARQUITETURA DO BLOCO T

O bloco T do codificador H.264/AVC reúne as três transformadas descritas anteriormente. Este bloco é responsável, também, pelo sincronismo dos dados de entrada e saída, pois os dados de entrada não são operados por todas as transformadas. A fig. 5 ilustra o diagrama em blocos para o bloco T desenvolvido neste trabalho.

A FDCT 4x4 é aplicada a todos os dados de entrada do bloco T. Após, existem três caminhos possíveis que dependem do tipo de dado e do modo de operação do codificador. Estes três caminhos podem ser observados na fig. 5. O primeiro caminho, composto pelo buffer Y_{DC} e a FHAD 4x4, é utilizado apenas no modo de operação INTRA 16x16. Neste modo, os elementos DC de luminância são processados pela FHAD 4x4 e os elementos AC são enviados diretamente para a quantização AC [2] pelo terceiro caminho. O caminho central, composto pelo buffer C_{DC} e pela FHAD 2x2, opera sobre os elementos DC de crominância e envia seus dados para a quantização DC [2].

Uma das principais vantagens da implementação paralela do bloco T é a drástica simplificação do controle. Na versão serial o controle é bastante complexo pois são necessários diversos buffers para realizar o sincronismo dos dados de entrada e manter a ordem correta dos dados de saída. Já na versão totalmente paralela, apenas três buffers são necessários: dois para sincronizar os dados de entrada das Hadamard 4x4 e 2x2 e um para acumular os elementos enviados pela Hadamard 2x2. Os resultados de saída do bloco T são disponibilizados em 3 saídas de 16 amostras. Cabe à quantização (bloco Q) realizar o controle dos dados de sua entrada. Esta estratégia é possível devido ao bloco de quantização possuir uma complexidade muito inferior a do bloco T e, também, por que a quantização ocupa um espaço muito menor no FPGA. Sendo assim, duas quantizações serão aplicadas aos dados de saída do bloco T, uma apenas para os elementos DC e outra apenas para os elementos AC.

5. RESULTADOS DE SÍNTESE

Este capítulo apresenta os resultados de síntese obtidos neste trabalho. Todas as arquiteturas desenvolvidas foram descritas em VHDL, utilizando a ferramenta ISE da Xilinx. A síntese das arquiteturas foi direcionada ao FPGA xc2vp70, da família Virtex II Pró da Xilinx [5]. Os principais resultados de síntese são apresentados na tab. 1.

Tabela 1 – Resultados de síntese

	LUTs	**Freqüência (MHz)**	**Throughput (Mamostras/s)**
FHAD 4x4	928	303,6	4.857,6
FDCT 4x4	656	319,7	5.115,2
FHAD 2x2	108	311,4	4.982,4
Controle	129	561,7	–
Bloco T	1783	303,6	4.857,6

Tabela 2 – Comparação entre o bloco T serial e paralelo

	LUTs	**Freqüência (MHz)**	**Throughput (Mamostras/s)**
Bloco T serial	2359	138,7	138,7
Bloco T paralelo	1783	303,6	4857,6

A tab. 1 omite os resultados de síntese para os buffers. No entanto, o bloco T reúne todas as arquiteturas da tab. 1 e os buffers de sincronização de dados apresentados na fig. 5.

O principal dado apresentado na tab. 1 é o throughput alcançado pelas arquiteturas. O bloco T gera dados a uma taxa de mais de 4,8 giga amostras por segundo, ou mais de 1561 quadros HDTV por segundo. Essa taxa possibilita, com larga margem de folga, o processamento de vídeos HDTV em tempo real (30 quadros por segundo). Esta margem de folga acentuada torna esta solução extremamente flexível, sendo útil para diferentes arquiteturas de codificadores H.264/AVC.

Outro aspecto importante de salientar é a validação das arquiteturas desenvolvidas. Para esta etapa foi utilizada a ferramenta Modelsim da Mentor Graphics. Foi desenvolvido um testbench que lê os arquivos com os dados de entrada e gera os sinais necessários para disparar o processamento no bloco T. Os resultados gerados são salvos em um arquivo de saída. Um software em linguagem C foi desenvolvido para comparar os resultados obtidos com os resultados esperados. Os valores usados na comparação foram extraídos do software de referência do padrão H.264/AVC.

6. RESULTADOS COMPARATIVOS

Existem poucos artigos na literatura que apresentam arquiteturas dedicadas para o bloco das transformadas do padrão H.264/AVC. Exceto por um trabalho anterior de nosso grupo [6], nenhuma outra implementação do bloco T completo foi encontrada na literatura. Os trabalhos publicados relativos ao bloco T, normalmente, tratam apenas de implementações em hardware de uma ou outra transformada deste bloco. Além disso, a maioria destes trabalhos é direcionada para tecnologia *stadard-cells*.

A primeira comparação realizada foi com o trabalho anterior de nosso grupo, que fez a implementação do bloco T de maneira serial e direcionada para FPGAs.

A tab. 2 apresenta os resultados de síntese para as implementações serial e paralela do bloco T.

Tabela 3 – Resultados Comparativos

Solução	Tec.	Nível de //	Nro de Gates	Throughput (Mamostras/s)
Agostini [10]	0.35µ	16	18,353	3,499
Nossa FDCT	0.35µ	16	6,440	3,048
Kordasiewicz [6]	-	16	77,280	1,720
Cheng [7]	0.35µ	8	5,745	800
Chen [8]	0.18µ	8	6,482	800
Wang [9]	0.35µ	4	6,538	320
Lin [11] FDCT	0.35µ	8	15,327	261
Agostini [5]	0.35µ	1	10,605	138

Da tab. 2 pode-se notar que o bloco T paralelo, além de possuir um *throughput* muito superior ao bloco T serial, utiliza um número menor de LUTs. Isto se deve, principalmente, pela redução drástica do número de buffers necessários para manter o sincronismo e, também, pela redução no tamanho do bloco de controle.

Para permitir uma comparação do trabalho apresentado neste artigo com outros trabalhos encontrados na literatura, foi realizada uma síntese do bloco da FDCT 4x4 direcionada para tecnologia *standard-cells* 0,35µm da TSMC. Este resultado, bem como os resultados de trabalhos relacionados, estão apresentados na tab. 3. Nesta tabela estão apresentadas a tecnologia utilizada, o nível e paralelismo de cada solução, o número de *gates* utilizados e o máximo *throughput* obtido em cada solução.

A solução apresentada em [6] realiza somente o cálculo da FDCT. As soluções [7], [8], [9] e [10] são arquiteturas multitransformadas que são aptas a processar os cálculos relacionados às quatro transformadas diretas e inversas de dimensões 4x4. A solução [10] também calcula a Hadamard 2x2 direta e inversa e permite a seleção no nível de paralelismo dos cálculos. Em todos os casos, apenas uma transformada é calculada a cada instante de tempo. As arquiteturas apresentadas em [11] agrupam, individualmente, cada transformada 4x4 com suas quantizações específicas, mas não apresenta um bloco T completo.

Considerando todas as soluções apresentadas na tab. 3, a solução apresentada neste trabalho é a que apresenta a melhor relação entre o número de *gates* utilizado e o *throughput* obtido, indicando o elevado grau de eficiência da arquitetura desenvolvida.

7. CONCLUSÕES

Este trabalho apresentou uma implementação em hardware para o bloco das transformadas diretas do padrão de compressão de vídeo H.264/AVC. Foram desenvolvidas versões totalmente paralelas para as três transformadas que compõem o bloco T (FDCT 4x4, FHAD 4x4 e FHAD 2x2). O bloco T foi desenvolvido utilizando-se estas transformadas totalmente paralelas e mais alguns buffers para a sincronização dos dados de entrada e de saída.

Com a arquitetura mapeada para FPGAs da Xilinx, a taxa de processamento obtida para o bloco T atinge de mais de 4,8 bilhões de amostras por segundo, permitindo o processamento de mais de 1561 quadros HDTV a cada segundo. Essa elevada taxa de processamento amplia a flexibilidade no uso desta solução em codificadores H.264/AVC focando HDTV em tempo real.

Exceto por outro trabalho de nosso grupo, não foi encontrada, na literatura, nenhuma outra solução para o bloco T completo do padrão H.264/AVC. Comparando apenas a FDCT 2-D 4x4 desenvolvida neste trabalho, com outras soluções publicadas na literatura, foi possível constatar que esta solução é a que apresenta um dos maiores *throughputs* e apresenta a melhor relação entre o uso de recursos de hardware e o *throughput* obtido.

8. REFERÊNCIAS

[1] Joint Video Team, "Draft ITU–T Recommendation and Final Draft International Standard of Joint Video Specification (ITU–T Rec. H.264)", 2003.

[2] Richardson, I, *H.264 and MPEG-4 Video Compression – Video Coding for Next-Generation Multimedia*, John Wiley and Sons, Chichester, 2003.

[3] G. Sullivan, and T. Wiegand, "Video Compression – From Concepts to the H.264/AVC Standard", *Proceedings of the IEEE*, IEEE, v. 93, n. 1, pp. 18-31, 2005.

[4] H. Malvar, et al, "Low-Complexity Transform and Quantization in H.264/AVC", *IEEE Trans. on Circuits and Systems for Video Tech.*, IEEE, v. 13, n. 7, pp. 598-603, 2003.

[5] L. Agostini, et all, "High Throughput Architecture for H.264/AVC Forward Transforms Block", *ACM Great Lake Symposium on VLSI*, Philadelphia, 2006.

[6] R. Kordasiewicz, and S. Shirani, "Hardware Implementation of the Optimized Transform and Quantization Blocks of H.264", *Canadian Conf. on Electrical and Computer Engineering*, Toronto, pp. 943-946, 2004.

[7] Z. Cheng, et al, "High Throughput 2-D Transform Architectures for H.264 Advanced Video Coders", *IEEE Asia-Pacific Conf. Circuits & Systems*, Taiwan,pp.1141-1144,2004.

[8] K. Chen, et al, "An Efficient Direct 2-D Transform Coding IP Design for MPEG-4 AVC/H.264", *IEEE International Symposium on Circuits and Systems*, Kobe, pp.4517-4520, 2005.

[9] T. Wang, et al, "Parallel 4x4 2D Transform and Inverse Transform Architecture for MPEG-4 AVC/H.264", *IEEE International Symposium on Circuits and Systems*, Bangkok, pp. II800-II803, 2003.

[10] L. Agostini, et all, "High Throughput Multitransform and Multiparallelism IP for H.264/AVC Video Compression Standard", *IEEE International Symposium on Circuits and Systems*, Kos Island, 2006.

[11] H. Lin, et al, "Combined 2-D Transform and Quantization Architectures for H.264 Video Coders", *IEEE International Symp. on Circuits and Systems*, Kobe, pp. 1802-1805, 2005.