

# PREDITOR DE VETORES DE MOVIMENTO PARA O PADRÃO H.264/AVC PERFIL MAIN

*Bruno Zatt<sup>1</sup>, Arnaldo Azevedo<sup>1</sup>, Luciano Agostini<sup>1,2</sup>, Sergio Bampi<sup>1</sup>*

<sup>1</sup> Grupo de Microeletrônica – II – UFRGS – Porto Alegre, Brasil

<sup>1,2</sup> Grupo de Arquiteturas e Circuitos Integrados – DInfo – UFPel – Pelotas, Brasil

{bzatt, apafilho, agostini, bampi}@inf.ufrgs.br

## RESUMO

Esse artigo apresenta uma arquitetura para o preditor de vetores de movimento do padrão H.264/AVC no perfil *Main*. O preditor de vetores de movimento é um dos módulos principais da compensação de movimento. A arquitetura foi desenvolvida para trabalhar a 100 MHz, fornecendo uma taxa de processamento capaz de decodificar HDTV em tempo real. O hardware foi modelado a partir de um banco de registradores e uma máquina de estados que opera sobre os dados registrados. O hardware foi sintetizado pra FPGA Xilinx Virtex II- PRO e *standard cells* utilizando tecnologia TSMC 0,18 $\mu$ m.

## 1. INTRODUÇÃO

O H.264/AVC ou MPEG-4 parte 10 – AVC (*Advanced Video Coding*) [1], [2] é o mais novo padrão de compressão de vídeo, desenvolvido em cooperação pela ITU-T *Video Coding Experts Group* (VCEG) e pelo ISO/IEC *Moving Picture Experts Group* (MPEG). O H.264/AVC, atingiu o objetivo de obter taxas mais elevadas de compressão de vídeo, chegando a reduzir em 50% a quantidade de bits necessários para representação de um vídeo, quando comparado a padrões anteriores. Esse aumento na capacidade de compressão, se deve a um considerável incremento na complexidade computacional dos codecs H.264/AVC em relação a padrões mais antigos. Com o aumento da complexidade computacional a codificação e decodificação de vídeos H.264/AVC se mostrou intratável por meio de software na tecnologia atual, impulsionando, assim o, desenvolvimento de codecs em hardware para esta tarefa.

Este artigo descreve uma arquitetura desenvolvida para efetuar a operação de predição de vetores de movimento segundo o padrão H.264/AVC em seu perfil *Main* [1].

A arquitetura proposta suporta todos os modos de predição de vetores presentes no H.264/AVC. A arquitetura foi construída de forma incremental, onde

primeiramente foi desenvolvida a predição padrão e, em seguida, da predição direta.

Este trabalho está inserido no projeto de um codec H.264/AVC em hardware para TV de alta definição (HDTV – 1920x1080). Este codec faz parte das investigações acadêmicas que estão sendo usadas de subsidio para a construção do Sistema Brasileiro de Televisão Digital (SBTVD).

O artigo trará, na seção 2, uma breve apresentação das características gerais e dos principais blocos contidos no padrão H.264/AVC. A seção 3 analisa a compensação de movimento no padrão. Na seção 4 é abordado o método de predição dos vetores de movimento. A seção 5 apresenta o projeto da arquitetura do preditor de vetores. A seção 6 contém os resultados e comparações. A seção 7 conclui o artigo.

## 2. PADRÃO H.264/AVC

O padrão H.264/AVC trabalha sobre macroblocos (MB) de 16x16 amostras e considerando imagens no espaço de cores YCbCr. Estes macroblocos são classificados de acordo com o modo de codificação ao qual são submetidos, podendo ser do tipo I (Intra), P (*Predictive*) e B (*Bi-predictive*). Os *slices*, estruturas que agrupam os macroblocos, também são classificados, podendo ser dos tipos I (Intra), P (*Predictive*), B (*Bi-predictive*), SP (*Switching P*) ou SI (*Switching I*).

Dentre as funcionalidades básicas de um codificador H.264/AVC estão as funções responsáveis pela decodificação de entropia, pelas transformadas inversas, pela quantização inversa, pela predição intraquadro, pela predição interquadro (MC) e pelo filtro de deblocação [1], como mostrado na Figura 1.

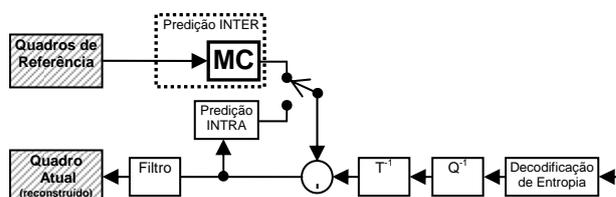


Figura 1. Decodificador H.264/AVC

### 3. COMPENSAÇÃO DE MOVIMENTO

A compensação de movimento é responsável por reconstruir macroblocos do quadro atual a partir de quadros vizinhos, chamados de quadros de referência.

Conhecendo os quadros de referência a serem utilizados, vetores de movimento indicam a localização das amostras desejadas. Estas amostras são interpoladas com uma precisão de 1/4 de pixel para luminância e 1/8 de pixel para crominância e, então, são somadas aos blocos de resíduo vindos das transformadas para reconstruir o macrobloco desejado.

Macroblocos P são preditos a partir de um único quadro de referência, enquanto macroblocos B podem utilizar informações de até dois quadros de referência diferentes. As referências são organizadas em duas listas, chamadas de **lista 0** e **lista 1**.

O bloco de compensação de movimento pode ser separado em dois módulos principais: o preditor de vetores e o processador de amostras. O preditor de vetores é responsável por calcular os vetores de movimento do bloco atual, utilizando dados do bitstream, além de informações contidas em blocos vizinhos. Essa operação é realizada para explorar a elevada correlação entre vetores de movimento de blocos vizinhos. O processador de amostras é responsável pela interpolação e pela predição ponderada.

Para atingir maiores taxas de compressão, o padrão H.264/AVC introduziu técnicas que não faziam parte de padrões anteriores devido a sua grande complexidade computacional. A compensação de movimento engloba a capacidade de referenciar múltiplos quadros, a predição ponderada, a predição direta, macroblocos *skip*, múltiplos tamanhos de bloco e interpolação de 1/4 pixel.

### 4. PREDITOR DE VETORES DE MOVIMENTO

Devido à elevada correlação entre blocos vizinhos, o padrão H.264/AVC define que os vetores de movimento, assim como os índices dos quadros de referência, sejam inferidos com base nos vetores de movimento e índices de referência dos blocos vizinhos. Esses vetores são chamados de vetores de movimento preditos (*predictive motion vectors* - PMV).

Desta forma, apenas os vetores de movimento diferenciados são codificados no *bitstream*, para depois serem somados aos PMVs, obtendo, assim, os vetores de movimento do bloco atual. Os PMVs são normalmente obtidos aplicando a mediana aos vetores dos blocos vizinhos. No entanto, macroblocos *skip* e macroblocos que sofrem predição direta são tratados de maneiras diferentes.

Esses vetores de movimento calculados apresentam precisão de 1/4 de amostra e são utilizados para interpolação de luminância. A partir desses vetores são derivados os vetores de movimento para crominância, com precisão de 1/8 de amostra. A ordem de predição segue a ordem de duplo Z ou zig-zag scan [1].

No H.264/AVC, uma importante inovação é o uso de blocos de tamanho variável, onde se pode usar uma partição do macrobloco em blocos de tamanho 16x16, 16x8, 8x16, 8x8, 4x8, 8x4 ou 4x4 para realizar a MC.

O bloco corrente é denominado bloco 'E'. As partições contíguas ao bloco corrente são nomeadas com as letras 'A', 'B', 'C' e 'D'. Sendo 'A' o bloco localizado a esquerda de 'E', 'B' é o bloco localizado imediatamente acima de 'E', 'C' é a partição acima e a direita de 'E' e, finalmente, 'D' se localiza acima e a esquerda do bloco corrente 'E' [1]. A Figura 2 exemplifica a relação entre os blocos vizinhos além dos múltiplos tamanhos de bloco possíveis.

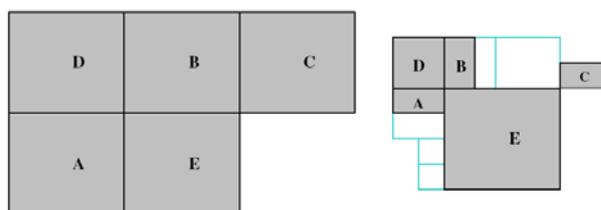


Figura 2. Relação entre blocos vizinhos (a) em formatos idênticos e (b) em formatos diferentes

Em *slices* P, é utilizado o modo de predição padrão, além de um modo específico para macroblocos *skip*. Para *slices* P a predição é procedida apenas para **lista 0**.

A predição padrão começa pela verificação da presença dos blocos vizinhos. Nesse estágio, é verificado se os blocos se encontram dentro do quadro, se pertencem ao *slice* corrente e se já foram decodificados. Blocos que não satisfazem essas condições são marcados com não disponíveis e seus vetores são considerados zero.

A operação padrão para o cálculo do PMV é a mediana aplicada a três vetores de movimento dos blocos vizinhos. Normalmente, são utilizados os vetores dos blocos 'A', 'B' e 'C'. Caso 'C' esteja não disponível, é utilizado o vetor do bloco 'D'. No caso de apenas um dos blocos ter a mesma referência do bloco corrente, seu vetor é definido como PMV.

Uma vez que o PMV é calculado, a próxima etapa é somar a ele o vetor de movimento diferencial presente no bitstream. O resultado é o vetor de movimento definitivo, que é atribuído a todo o bloco, seja ele um macrobloco, uma partição ou uma sub-partição.

A predição dos vetores de movimento em *slices* tipo B pode ser efetuada através do modo padrão ou utilizando predição direta. A predição direta se divide em direta espacial e direta temporal. A predição direta espacial é uma inovação do H.264/AVC, enquanto a predição direta temporal é uma simplificação da predição encontrada no MPEG-4 parte 2 [2].

A predição direta utiliza informações presentes no quadro co-localizado que, segundo o padrão H.264/AVC, é o quadro de referência índice 0 da **lista 1**. Quando se fala em bloco co-localizado, se está referindo ao bloco do quadro co-localizado que se encontra na mesma posição que o bloco corrente.

A predição direta espacial inicia o processo de decodificação calculando os índices dos quadros de referência. Para isso, é verificada a presença das partições e sub-partições vizinhas. Caso estas estejam não disponíveis, seu índice de referência assume o valor zero. Assim, como na predição padrão, caso ‘C’ não esteja disponível, o valor de ‘D’ é utilizado.

Após encontrar os valores dos índices de referência da vizinhança, uma operação chamada “Menor Positivo” é aplicada sobre os índices de referência para determinar o valor da referência do macrobloco atual. Essa operação assume o valor do menor índice de referência positivo. Caso todos os índices sejam negativos, assume-se o maior índice negativo.

Nesta fase também são determinados os vetores de movimento com base nos vetores dos blocos vizinhos ao macrobloco corrente.

A outra possibilidade de predição de vetores em *slices* tipo B, a predição direta temporal, é uma simplificação da predição utilizada pelo MPEG-4 Parte 2. O vetor é calculado com base no vetor co-localizado e na distância entre o quadro atual e os quadros de referência. A distância entre os quadros é calculada utilizando o contador de ordem de imagem (POC – *Picture Order Counter*) [1].

Um fator de escala é determinado a partir da diferença entre o POC do quadro de referência da **lista 0** e o POC do quadro de referência **lista 1**.

O fator de escala é aplicado ao vetor de movimento do bloco co-localizado para obter o valor do vetor para **lista 0**. O vetor para **lista 1** é a subtração do vetor da **lista 0** do vetor co-localizado.

## 5. ARQUITETURA DO MVP

O preditor de vetores foi desenvolvido para trabalhar em conjunto com o filtro para compensação de movimento apresentado em [3] tendo como objetivo principal implementar um compensador de movimento completo para H.264/AVC no perfil *Main*. Visando atingir desempenho para decodificar HDTV em resolução 1920x1080 pixels e amostragem 4:2:0, foi determinado que o módulo como um todo deveria trabalhar a uma frequência de 100 MHz, fornecendo uma amostra processada por ciclo de relógio. Uma vez que existem 256 amostras de luminância e 128 amostras de crominância em um macrobloco, o preditor de vetores tem como restrição temporal decodificar um macrobloco em, no máximo, 384 ciclos de relógio para não limitar a capacidade de processamento do filtro de amostras.

A arquitetura foi descrita em VHDL utilizando uma descrição mista entre VHDL estrutural e comportamental.

Devido à necessidade de consultar informações dos blocos vizinhos, é necessário que todos macroblocos da linha superior sejam armazenados pela arquitetura. Foram, então, descritos dois buffers em memória, um para armazenar os vetores e outro para armazenar os índices de referência dos blocos vizinhos. Como essa

implementação visa decodificar HDTV com 1920 pixels de largura, o buffer que armazena os vetores de movimento deve armazenar 480 vetores. Para isso, foi dimensionado em 512 palavras de 32 bits. O buffer de índices de referência deve armazenar 240 índices e, portanto, foi dimensionado em 60 palavras de 32 bits. Foram utilizadas palavras de 32 bits devido à largura das palavras da memória RAM interna ao FPGA.

A arquitetura também conta com 10 conjuntos de registradores para o armazenamento dos vetores e índices vizinhos e internos do macrobloco corrente. A Figura 3 representa a disposição dos conjuntos de registradores que armazenam os vetores de movimento. As caixas com letras, na Figura 3, representam os vizinhos a qual armazenam os dados. As caixas com fundo em cinza representam os registradores que armazenam os vetores de movimento do próprio macrobloco. Dois conjuntos com 4 registradores armazenam os vetores de movimento imediatamente à esquerda e acima do macrobloco corrente. Esses vetores são referentes aos vizinhos “A” e “B” do macrobloco corrente, respectivamente. Dois registradores armazenam os vetores do bloco vizinho superior esquerdo e do superior direito, referentes aos vizinhos “D” e “C”, respectivamente. Um conjunto de 16 registradores armazena os vetores de movimento dos blocos do macrobloco corrente.

Os outros cinco conjuntos de registradores armazenam os índices de referência dos blocos vizinhos.

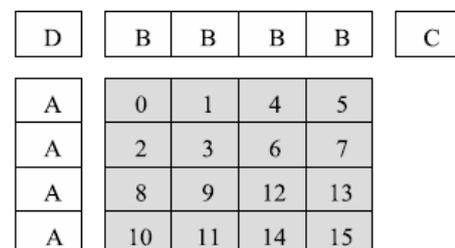


Figura 3. Estrutura de registradores para armazenamento de vetores internos ao macrobloco (0 a 15) e vetores de macroblocos vizinhos (A, B, C e D)

As operações sobre esses conjuntos de registradores foram descritas na forma de uma máquina de estados. Essa máquina de Mealy deve processar um macrobloco em, no máximo, 384 ciclos de relógio. Cada macrobloco pode ser dividido em até 16 blocos 4x4, que são processados de forma serial pela máquina de estados. Para tanto, cada bloco deve ser processado em até 24 ciclos de relógio.

O hardware foi desenvolvido de maneira incremental. Primeiramente foi desenvolvida a arquitetura para predição padrão para *slices* tipo P, seguido da descrição para predição direta espacial e, finalmente, o desenvolvimento da predição direta temporal.

Em sua versão final, essa arquitetura conta com uma máquina de estados com 50 estados, tem latência mínima

de 17 ciclos de clock e uma latência máxima de 226 ciclos de clock. Somando-se os 16 ciclos finais para envio de resultados, a arquitetura completa gasta, no máximo, 242 ciclos de clock para processar um macrobloco, mantendo-se dentro dos 384 ciclos disponíveis para esse processo.

Além da máquina de estados, foi criada uma arquitetura específica para calcular o fator de escala utilizado na predição direta temporal. Essa arquitetura foi implementada em um *pipeline* de 3 estágios. O primeiro estágio é composto por dois subtratores de 18 bits, dois operadores de *clip* (utilizados para manter as saídas dos subtratores no intervalo [0, 255]) e um somador que adiciona a constante 16384 ao resultado de uma das subtrações. No segundo estágio existe um divisor de 16 bits. O último estágio conta com um multiplicador 18 bits, um somador com a constante 32, um deslocador (*shifter*) e um operador de *clip*.

## 6. RESULTADOS E COMPARAÇÕES

Os dados de síntese apresentados nessa seção foram gerados pelo *software* ISE da Xilinx [4]. Os dispositivos alvo dessa síntese foram FPGAs VP-30 da família Virtex-II PRO da Xilinx com *speed grade -7* (xc2vp30-7ff896).

A Tabela 1 apresenta os dados de utilização dos recursos do dispositivo, considerando número absoluto e porcentagem de utilização. A frequência máxima de operação também é apresentada.

Tabela 1. Resultados de Síntese para FPGA

<b>LUTs</b>	6.221 (22%)
<b>Registers</b>	3.596 (13%)
<b>Multipliers</b>	3 (2%)
<b>Clock</b>	133,9 MHz

O preditor de vetores de movimento também foi sintetizado em *standard cells*, utilizando tecnologia TSMC 0,18 $\mu$ m. Os resultados gerados são mostrados na Tabela 2. O software utilizado para síntese foi o Leonardo Spectrum da Mentor Graphics [5].

Tabela 2. Resultados de Síntese para Standard Cells

<b>Gate Count</b>	46.886
<b>Clock</b>	129,3 MHz

A frequência de operação mais baixa em *standard cells* com relação ao FPGA se deve ao alto desempenho dos multiplicadores implementados fisicamente no FPGA.

A arquitetura mapeada para o FPGA alvo é capaz de processar 185 quadros HDTV por segundo, enquanto a arquitetura em *standard-cells* atinge uma taxa de processamento de 179 quadros HDTV por segundo. Em ambos os casos, a arquitetura desenvolvida para o MVP suporta decodificar quadros HDTV em tempo real.

Não foi encontrada, na literatura, nenhuma implementação completa desse módulo em hardware para o perfil *Main* do padrão H.264/AVC. Em [6], esse módulo do compensador de movimento foi implementada em software. No artigo [7] é implementado um preditor com limitações. No artigo [8], um preditor de vetores completo foi implementado em VHDL, no entanto focando apenas o perfil *Baseline*.

## 7. CONCLUSÃO

Este artigo apresentou uma arquitetura para o preditor de vetores de movimento do padrão H.264/AVC, capaz de decodificar HDTV em tempo real a 30 quadros por segundo. Nenhuma arquitetura equivalente, suportando todos os modos de predição do perfil *Main*, foi encontrada na literatura. Resultados para FPGA e *standard-cells* foram apresentados e atingem o desempenho acima do necessário para processar HDTV.

A arquitetura é baseada em um banco de registradores e uma máquina de estados que opera sobre esses dados.

Atualmente, a arquitetura desenvolvida está sendo prototipada em uma placa com o FPGA alvo. Como trabalhos futuros pretende-se concluir a prototipação e integrá-lo ao filtro de amostras para compensação de movimento apresentada em [3].

## 8. REFERÊNCIAS

- [1] JVT Editors (T. Wiegand, G. Sullivan, A. Luthra), Draft ITU-T Recommendation and final draft international standard of joint video specification (ITU-T Rec.H.264 ISO/IEC 14496-10 AVC), JVTG050r1, Geneva, May 2003.
- [2] T. Wiegand; G. Sullivan; G. J. Bjntegaard; A. Luthra; "Overview of the H.264/AVC video coding standard". Circuits and Systems for Video Technology, IEEE Transactions on. Volume 13, Issue 7, July 2003 Page(s):560 – 576
- [3] A. Azevedo, B. Zatt, L. Agostini, and S. Bampi, "Motion compensation sample processing for hdtv h.264/avc decoder" in IEEE Norchip, Nov. 2005, pp. 110-113.
- [4] Xilinx Inc.. Disponível em: <<http://www.xilinx.com/>>, 2006
- [5] Mentor Graphics. LeonardoSpectrum. Disponível em: [http://www.mentor.com/products/fpga\\_pld/synthesis/leonardo\\_spectrum/](http://www.mentor.com/products/fpga_pld/synthesis/leonardo_spectrum/), 2006.
- [6] Shih-Hao Wang, and Tihao Chiang et al., "A platformbased MPEG-4 advanced video coding (AVC) decoder with block level pipelining," ICICS-PSM, December 2003.
- [7] Tung-Chien Chen, Yu-Wen Huang, and Liang-Gee Chen, "Fully utilized and reusable architecture for fractional motion estimation of H.264/AVC," IEEE International Conference on Acoustics, Speech, and Signal Processing, May 2004.
- [8] Sheng-Zen Wang, Ting-An Lin, Tsu-Ming Liu and Chen-Yi Lee. A New Motion Compensation Design for H.264/AVC Decoder. In: Isacas 2005 Ieee International Symposium On Circuits And Systems, 2005.