

## ARQUITETURA EM HARDWARE PARA A UNIDADE DE INTERPOLAÇÃO DO PADRÃO HEVC DE CODIFICAÇÃO DE VÍDEO

**CORRÊA, Marcel<sup>1</sup>; AFONSO, Vladimir<sup>2</sup>; JUNES, Victor<sup>1</sup>; AGOSTINI, Luciano<sup>3</sup>**

<sup>1</sup>Universidade Federal de Pelotas, Curso de Ciência da Computação.  
e-mail: mmcorrea@inf.ufpel.edu.br, vrcjunes@inf.ufpel.edu.br

<sup>2</sup>Universidade Federal de Pelotas, Programa de Pós-Graduação em Computação.  
e-mail: vafonso@inf.ufpel.edu.br

<sup>3</sup>Universidade Federal de Pelotas, Centro de Desenvolvimento Tecnológico.  
e-mail: agostini@inf.ufpel.edu.br

### 1 INTRODUÇÃO

A codificação de vídeo é uma importante área de pesquisa nos dias de hoje em função do aumento acelerado da demanda por vídeos digitais de alta definição. Existem diversos padrões de codificação de vídeo e estes definem duas coisas: (1) uma representação codificada (sintaxe), que descreve a informação visual de forma comprimida, e (2) um método de interpretar esta sintaxe para reconstruir a informação visual (RICHARDSON, 2003).

Atualmente, o padrão de codificação de vídeo mais eficiente disponível é o H.264/AVC (*Advanced Video Coding*) (JVT, 2003). Porém, as resoluções de vídeo e a quantidade de informação que precisam ser processadas estão aumentando com o passar do tempo. Considerando este aumento da demanda, um grupo de especialistas na área de codificação de vídeo chamado JCT-VC (*Joint Collaborative Team on Video Coding*) está desenvolvendo um novo padrão, projetado para ser mais eficiente que o antecessor H.264/AVC. Este novo padrão é atualmente chamado de HEVC (*High Efficiency Video Coding*) (JCT-VC, 2012) e representa o estado da arte em algoritmos para codificação de vídeo.

Um dos fatores que devem ser levados em consideração em um codificador é a complexidade computacional, uma vez que o processamento de vídeos de altíssima definição em tempo real requer que seja realizada uma quantidade muito elevada de cálculos, e nem mesmo os mais modernos processadores de propósito geral conseguem vencer esta restrição. Além disso, soluções deste tipo não são apropriadas para sistemas embarcados devido ao elevado consumo de energia. Logo, apesar de este ser um problema geral, ele é muito mais crítico em dispositivos móveis que são alimentados por uma bateria e, portanto, processadores de propósito específico ou circuitos integrados dedicados com elevada performance e baixo consumo de energia devem ser desenvolvidos para que estes dispositivos possam dar suporte ao padrão HEVC.

Codificadores de vídeo possuem vários módulos que exploram diferentes redundâncias na informação visual. Este trabalho tem como foco desenvolver uma arquitetura em hardware para parte do módulo da estimação de movimento (ME – *Motion Estimation*), o qual é responsável pelos melhores resultados em termos de qualidade final de compressão, mas que também é o mais complexo computacionalmente (RICHARDSON, 2003). A ME explora e reduz a redundância entre quadros próximos no tempo e funciona dividindo o quadro atual em vários blocos e procurando em quadros previamente codificados pelo bloco mais similar ao

que se está processando. Depois desta busca, um vetor indicando deslocamento entre os blocos é gerado.

Em geral, a ME é realizada para fazer uma busca de pixels inteiros. No entanto, é permitido também o uso de uma busca fracionária com precisão de até um quarto de pixel. Porém, para gerar os pixels fracionários, é necessário interpolar uma área ao redor dos pixels inteiros, ocorrendo assim uma nova etapa chamada interpolação de sub-pixels.

## 2 METODOLOGIA (MATERIAL E MÉTODOS)

O padrão HEVC propõe novos filtros de interpolação de sub-pixels, mais eficientes que aqueles utilizados no padrão H.264/AVC. No AVC, dois passos dependentes são utilizados para gerar amostras com precisão de um quarto de pixel. No HEVC, um único passo é realizado para obter estas amostras, aumentando assim o paralelismo do processo. Neste padrão mais recente, o método de interpolação utilizado é o filtro DCT (Transformada Discreta de Cosseno) de *8-taps*.

A arquitetura em hardware desenvolvida neste trabalho foi baseada no documento de definição (JCT-VC, 2012) e no software de referência do padrão HEVC (HM, 2012). Nestas referências é possível encontrar todos os procedimentos utilizados na interpolação de sub-pixels. A Figura 1 representa amostras inteiras (blocos sombreados) e amostras de posições fracionárias (blocos brancos). Esta figura, as equações (1-2) que são utilizadas para obter amostras fracionárias e a Tabela 1 são importantes para que se entenda o processo de interpolação.

$A_{-1,-1}$				$A_{0,-1}$	$a_{0,-1}$	$b_{0,-1}$	$c_{0,-1}$	$A_{1,-1}$				$A_{2,-1}$
$A_{-1,0}$				$A_{0,0}$	$a_{0,0}$	$b_{0,0}$	$c_{0,0}$	$A_{1,0}$				$A_{2,0}$
$d_{-1,0}$				$d_{0,0}$	$e_{0,0}$	$f_{0,0}$	$g_{0,0}$	$d_{1,0}$				$d_{2,0}$
$h_{-1,0}$				$h_{0,0}$	$i_{0,0}$	$j_{0,0}$	$k_{0,0}$	$h_{1,0}$				$h_{2,0}$
$n_{-1,0}$				$n_{0,0}$	$p_{0,0}$	$q_{0,0}$	$r_{0,0}$	$n_{1,0}$				$n_{2,0}$
$A_{-1,1}$				$A_{0,1}$	$a_{0,1}$	$b_{0,1}$	$c_{0,1}$	$A_{1,1}$				$A_{2,1}$
$A_{-1,2}$				$A_{0,2}$	$a_{0,2}$	$b_{0,2}$	$c_{0,2}$	$A_{1,2}$				$A_{2,2}$

Figura 1 – Amostras de posições inteiras e fracionárias.

$$Clip_{0-255} \left[ (C_1 A_{-3,0} + C_2 A_{-2,0} + C_3 A_{-1,0} + C_4 A_{0,0} + C_5 A_{1,0} + C_6 A_{2,0} + C_7 A_{3,0} + C_8 A_{4,0}) \gg 6 \right] \quad (1)$$

$$Clip_{0-255} \left[ (C_1 A_{0,-3} + C_2 A_{0,-2} + C_3 A_{0,-1} + C_4 A_{0,0} + C_5 A_{0,1} + C_6 A_{0,2} + C_7 A_{0,3} + C_8 A_{0,4}) \gg 6 \right] \quad (2)$$

Tabela 1 – Coeficientes utilizados nos filtros.

Tipo	Posições	Coeficientes ( $C_2 \dots C_7$ )
L (esquerda)	$a_{i,j}, d_{i,j}, e_{i,j}, f_{i,j}, g_{i,j}$	$\{-1, 4, -10, 58, 17, -5, 1, 0\}$
M (meio)	$b_{i,j}, h_{i,j}, i_{i,j}, j_{i,j}, k_{i,j}$	$\{-1, 4, -11, 40, 40, -11, 4, -1\}$
R (direita)	$c_{i,j}, n_{i,j}, p_{i,j}, q_{i,j}, r_{i,j}$	$\{0, 1, -5, 17, 58, -10, 4, -1\}$

A equação (1) é utilizada para calcular amostras fracionárias que estão alinhadas horizontalmente entre posições inteiras (Tipo H:  $a_{0,0}, b_{0,0}, c_{0,0}$ ), e a equação (2) é utilizada para amostras alinhadas verticalmente entre posições inteiras (Tipo V:  $d_{0,0}, h_{0,0}, n_{0,0}$ ). As variáveis  $C_1$  até  $C_8$  são coeficientes que se alteram de acordo com a posição do sub-pixel.

Para calcular aqueles sub-pixels que não estão alinhados nem horizontalmente e nem verticalmente entre amostras inteiras (Tipo D), são utilizadas as amostras calculadas através da equação (1) ao invés das amostras de posição inteira. Para isso, utiliza-se a equação (2).

No trabalho desenvolvido, blocos de tamanho  $8 \times 8$  pixels foram considerados e três filtros diferentes foram projetados, um para cada grupo de coeficientes, de acordo com a Tabela 1. Estes filtros possuem uma etapa final chamada de *clip*, como pode ser visto nas equações (1-2), que limita o resultado dentro da faixa de 0-255 (mantendo a precisão de 8 bits). As multiplicações presentes nestas equações foram convertidas para somas e deslocamentos, para obtenção de uma arquitetura mais eficiente. Além dos filtros descritos na parte operativa, bancos de registradores foram desenvolvidos com a capacidade de armazenar e deslocar linhas ou colunas inteiras de dados. Desta forma, estes filtros são capazes de receber e calcular uma linha ou coluna inteira de amostras em um único passo. A Figura 2 mostra o diagrama de blocos simplificado desta arquitetura.

Em relação ao desempenho, são necessários: 16 ciclos de *clock* para processar e armazenar amostras alinhadas horizontalmente entre amostras inteiras, 8 ciclos para amostras alinhadas verticalmente, e 27 ciclos para amostras não alinhadas. Sendo assim, são necessários 51 ciclos de *clock* para processar um bloco de tamanho  $8 \times 8$ .

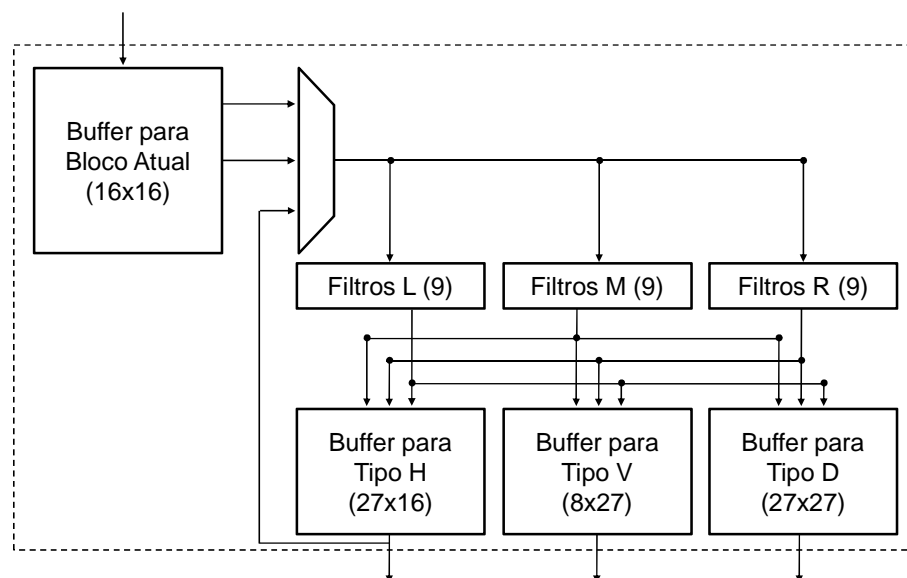


Figura 2 – Diagrama de blocos simplificado com sinais de controle omitidos.

### 3 RESULTADOS E DISCUSSÃO

A arquitetura proposta foi descrita na linguagem VHDL de descrição de hardware e sintetizada para um dispositivo FPGA Xilinx Virtex 4 XC4VLX15 utilizando-se a ferramenta de síntese Xilinx ISE 10.1.

Como mostra a Figura 2, são utilizados um total de 27 filtros de interpolação, sendo 9 de cada tipo, para que se possa atingir o poder computacional necessário para processar vídeos de altíssima resolução em tempo real de processamento (30 quadros por segundo ou mais). Também foram utilizados bancos de registradores, para realizar a tarefa de armazenamento e deslocamento dos dados.

Quando sintetizada para o dispositivo XC4VLX15, a arquitetura atingiu uma frequência máxima de operação de 210 MHz (210 milhões de ciclos de *clock* por segundo). Como são necessários 51 ciclos para processar um bloco de tamanho 8x8, essa frequência permite que a arquitetura proposta processe 30 quadros QFHD (3840x2160 pixels) por segundo, ou 120 quadros FullHD (1920x1080) por segundo.

O bloco operativo, composto pelos filtros, consumiu 3267 *slice LUTs* (17%) do dispositivo FPGA. Separadamente, o banco de registradores utilizou 1633 registradores de 8 bits.

### 4 CONCLUSÃO

Neste artigo, uma arquitetura em hardware de alta performance para a unidade de interpolação de sub-pixels do padrão HEVC de codificação de vídeo foi apresentada.

Quando sintetizada para um dispositivo FPGA Xilinx Virtex 4, esta arquitetura atingiu um poder computacional muito elevado. Ela é capaz de processar vídeos de altíssima resolução, como QFHD, em tempo real. Ela também é capaz de processar vídeos de resolução FullHD executando em uma frequência mais baixa, o que é muito importante quando restrições de consumo de energia devem ser consideradas.

Como trabalho futuro, é planejado o desenvolvimento de uma arquitetura de ME totalmente funcional para o padrão HEVC, que suporte a precisão de um quarto de pixel e que utilize um algoritmo de busca rápido.

### 5 REFERÊNCIAS

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

JVT 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.

JCT-VC. High Efficiency Video Coding (HEVC) Text Specification Draft 6. Disponível em <<http://phenix.int-evry.fr/jct/>>. Acesso em: Abril de 2012.

HM 6.0: HEVC Test Model Reference Software. Disponível em: <<http://hevc.info/HM-doc/>>. Acesso em: Abril de 2012.