

# Avaliação do Custo de Comunicação com a Memória Externa de uma Arquitetura em Hardware para Estimação de Movimento H.264

Alba S. B. Lopes<sup>1</sup>, Ivan Saraiva Silva<sup>2</sup>

<sup>1</sup>Departamento de Informática e Matemática Aplicada  
Universidade Federal do Rio Grande do Norte (UFRN) – Natal, RN – Brasil

<sup>2</sup>Departamento de Informática e Estatística  
Universidade Federal do Piauí (UFPI) – Teresina, PI – Brasil

alba@lasic.ufrn.br, ivan@ufpi.edu.br

**Abstract.** *This paper presents an architecture for H.264 Motion Estimation. The proposed architecture uses a fixed block size 8x8 pixels and a search area 32x32. This paper also presents a cost evaluation of the communication of the proposed architecture with external memory using the Altera DE2 prototyping kit. The impact of this cost in architecture global performance is analyzed and some ideas to improve this performance are given.*

**Resumo.** *Este trabalho apresenta uma proposta de arquitetura para um núcleo de Estimação de Movimento segundo o padrão H.264. A arquitetura proposta utiliza um tamanho de bloco fixo 8x8 pixels e uma área de pesquisa 32x32. Neste trabalho é também apresentando um estudo realizado sobre a comunicação da arquitetura proposta com a memória externa utilizando o kit de prototipagem Altera DE2. É feita uma avaliação do impacto desta comunicação no desempenho final da arquitetura e são apresentadas propostas para melhorar este desempenho.*

## 1. Introdução

Atualmente, diversos aparelhos eletrônicos como celulares, DVD *players*, filmadoras, câmeras e TVs digitais são capazes de manipular vídeos digitais [Agostini 2007]. Um vídeo tal como foi capturado exige uma quantidade muito elevada de bits para a sua representação. Entretanto, uma característica intrínseca destes dados é a redundância que eles apresentam [Ghanbari 2003]. Dessa forma, grande parte dos dados utilizados para a representação da informação é desnecessária. A compressão de vídeos torna-se, então, essencial para possibilitar o armazenamento e principalmente a transmissão desses dados.

O padrão H.264 [ITU-T 2007] é o mais novo padrão de compressão de vídeo que atingiu seu objetivo ao reduzir em aproximadamente 50% a quantidade de bits necessária para a representação da informação ao ser comparado aos demais padrões existentes na atualidade. O ganho foi em consequência de um alto grau de complexidade computacional inserido na codificação.

Um vídeo é formado por uma seqüência de imagens, chamadas quadros, que costumam apresentar certo grau de similaridade entre eles. Esta similaridade é

conhecida como redundância temporal [Shi e Sun 1999]. No codificador de vídeo, o módulo responsável por tratar desta redundância é a Estimação de Movimento (do inglês *Motion Estimation - ME*). A ME é o módulo que possui a mais elevada complexidade computacional do codificador do padrão H.264 [Deng, Gao, Hu e Ji 2005]

Neste trabalho é apresentada uma arquitetura para um núcleo de ME e é feita uma avaliação do impacto da comunicação do núcleo com a memória externa à arquitetura. Na memória externa são armazenados os quadros do vídeo que devem ser codificados. Este é um assunto pouco abordado quando se trata de núcleos para a ME.

Este artigo está organizado da seguinte forma: a seção 2 apresenta uma breve descrição do funcionamento da ME. Na seção 3 é apresentado o núcleo para a estimação de movimento proposto. A seção 4 traz uma avaliação do desempenho do núcleo. A seção 5 apresenta um estudo de caso realizado com um kit de prototipagem que possui duas memórias externas disponíveis e apresenta o impacto no desempenho da arquitetura ao acrescentar os acessos à memória externa. Por fim, a seção 6 apresenta as conclusões e trabalhos futuros.

## 2. Estimação de Movimento

Para a estimação de movimento são fornecidos um quadro atual e quadros de referência (o padrão H.264 permite a utilização de múltiplos quadros de referência). Cada quadro atual é dividido em regiões não sobrepostas chamadas blocos [Verma e Akoglu 2008]. Para cada bloco, a ME deve realizar uma busca em uma região do quadro de referência denominada área de pesquisa, como apresentado na Figura 1. Esta busca deve resultar na geração de um vetor de movimento que indique o deslocamento relativo do bloco do quadro atual para a região do quadro de referência.

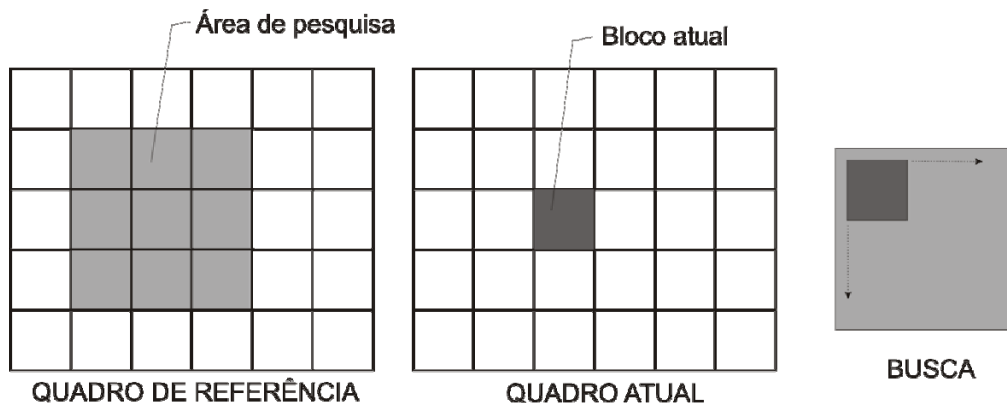


Figura 1. Representação da Estimação de Movimento

Existem diversos algoritmos para a busca que visam balancear desempenho e qualidade do vetor gerado. A Busca Completa (do inglês *Full Search*) prioriza a qualidade em relação ao desempenho [Bhaskaran e Konstantinides 1997]. Este algoritmo é chamado algoritmo ótimo, pois sempre encontra o melhor vetor existente. Algoritmos subótimos, como a Busca em Diamante (do inglês *Diamond Search*), fornecem um bom desempenho e um vetor aceitável, mas não o melhor [Yi e Ling 2005].

A avaliação da diferença entre dois blocos é feita através um critério de similaridade. Existem diferentes funções que servem para quantificar esta similaridade. Na ME, a medida mais utilizada é o SAD (do inglês *Sum of Absolute Differences*). [Vanne 2006] que calcula a diferença em módulo entre os pixels do bloco do quadro atual e seu respectivo no quadro de referência.

A arquitetura para a ME proposta neste trabalho utiliza tamanho de bloco 8x8 pixels, área de pesquisa 32x32, algoritmo da Busca Completa para a varredura da área de pesquisa e o SAD como critério de similaridade.

### 3. Arquitetura de um núcleo para a Estimação de Movimento

A arquitetura apresentada na Figura 2 é formada por uma memória local, três módulos de processamento, um banco de registradores de bloco e um comparador. Mais detalhes sobre cada um desses módulos serão apresentados nas subseções seguintes.

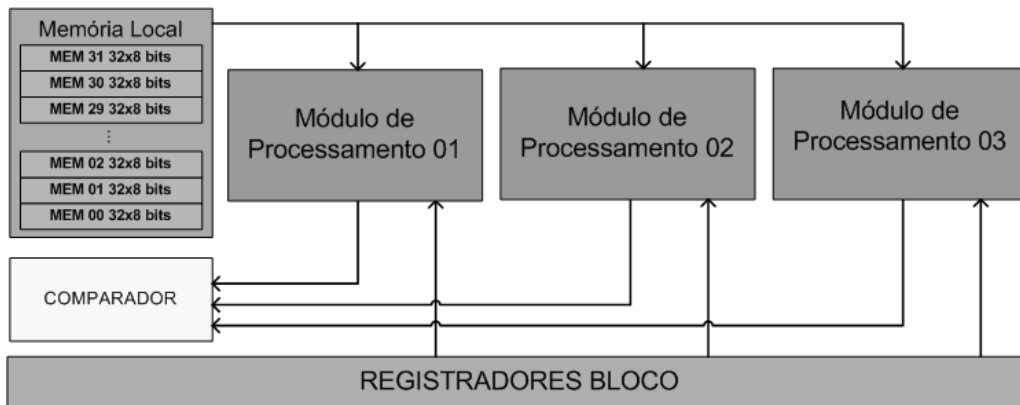


Figura 2. Arquitetura proposta de um núcleo para a Estimação de Movimento

#### 3.1 Banco de Registradores de Bloco

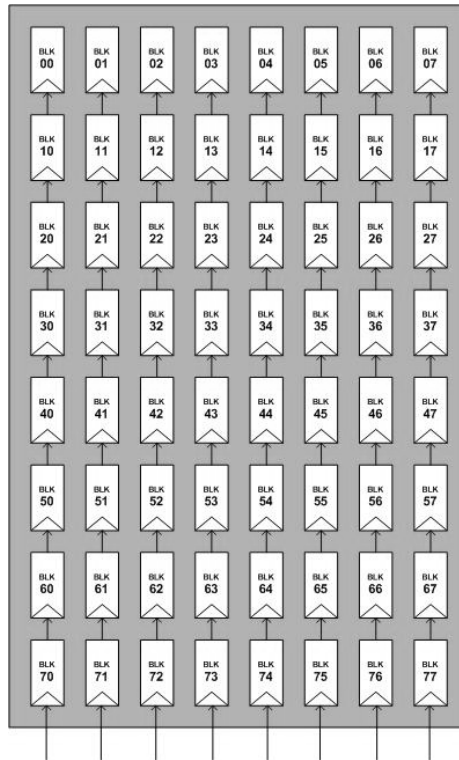
Este módulo possui 64 registradores (dispostos em forma de matriz quadrada na Figura 3) para armazenar os 64 valores referentes a um bloco 8x8. Cada registrador do módulo possui capacidade de armazenamento de 8 bits, referentes a 1 pixel do bloco. Oito pixels são passados como entrada para o módulo, que através de deslocamentos sucessivos realiza o preenchimento completo da matriz 8x8. Só há um banco de registradores de bloco no núcleo e todos os módulos de processamento utilizam os dados nele contidos.

#### 3.2 Módulo de Processamento

Cada Módulo de Processamento (MP) é formado por quatro Unidades de Processamento (UPs) e um banco de registradores de referência, como ilustra a Figura 4.a.

No banco de registradores de referência (Figura 4. b) são armazenados os valores de parte da área de pesquisa com a qual no momento será realizado o casamento com o bloco atual. Este banco é capaz de realizar deslocamento dos dados em todas as direções. Como exemplo, seja o dado que se encontra no registrador REF 00 representado na Figura 4.b. Após ser realizado um deslocamento para baixo, este dado irá para o registrador REF 10. Ao realizar um deslocamento para a direita, o dado

passará para o registrador REF 01. Em deslocamentos para cima e para a esquerda, o dado será descartado. O procedimento é análogo para os demais registradores do módulo.



**Figura 3. Representação do banco de registradores de bloco**

Tal mecanismo foi idealizado para que os dados contidos nos registradores possam ser reaproveitados em outro casamento, pois na Busca Completa, o bloco é deslocado pixel em pixel na área de pesquisa, varrendo todos os blocos candidatos [Rosa 2007]. Dessa forma, a diferença entre um casamento e outro é de apenas uma coluna/linha de pixel. Sendo assim, apenas os dados que ainda não estão disponíveis para o próximo casamento são lidos da memória e carregados nos registradores. Este procedimento é controlado por meio de um seletor que indica qual deslocamento os dados devem realizar em cada momento.

A arquitetura conta com 3 Módulos de Processamento. Esta quantidade de módulos foi escolhida para realizar o melhor reaproveitamento de dados que arquitetura permite e reduzir a quantidade de área necessária. Os casamentos entre o bloco e a área de pesquisa são distribuídos entre os 3 MPs. Neste caso, dado que a área de pesquisa possui tamanho 32x32 e o bloco 8x8, são necessários 625 casamentos para determinar o melhor vetor de movimento, utilizando a Busca Completa.

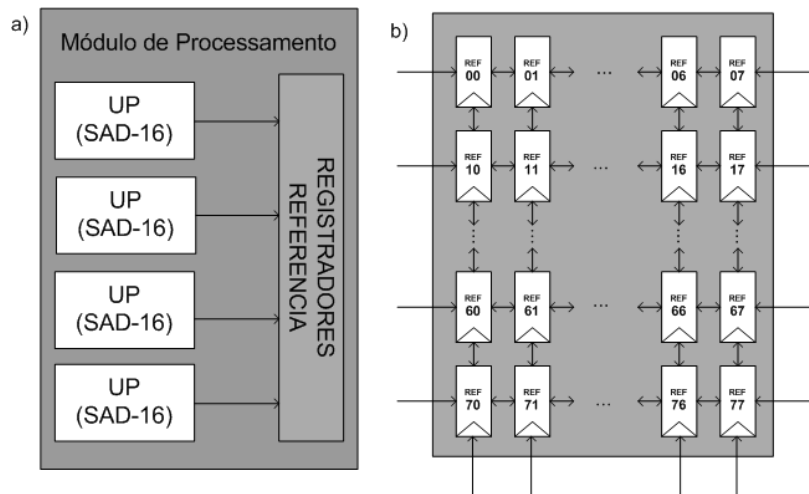


Figura 4. a) Detalhamento do Módulo de Processamento. b) Banco de registradores de referência.

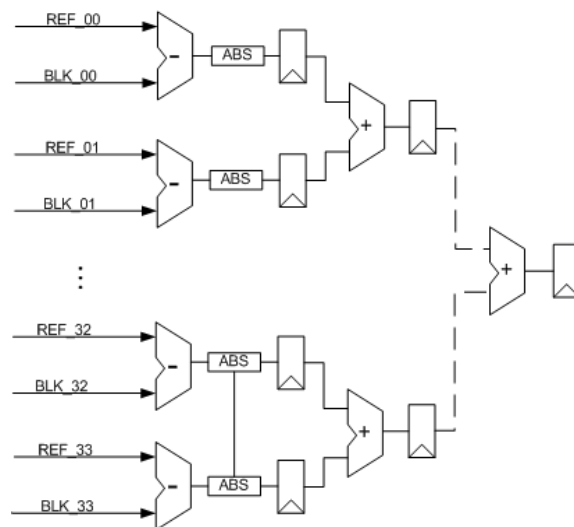


Figura 5. Detalhamento arquitetural da Unidade de Processamento

Destes 625 casamentos, o MP1 realiza 200, o MP2 realiza outros 200 e o MP3 realiza 225. Esta distribuição foi determinada tendo em vista o melhor reuso de dados que a arquitetura permite. Optou-se por deixar a MP3 com um pequeno número a mais de casamentos visando economia de área ocasionada pelo não acréscimo de outro MP à arquitetura para a realização dos 25 casamentos excedentes. Uma vez que cada casamento é realizado em um ciclo, isto acarreta uma ociosidade de 25 ciclos das duas outras MPs. Entretanto, esta ociosidade não se apresenta como um fator crítico tendo em vista a necessidade de preenchimento das memórias para o início de uma nova etapa de cálculos de similaridade. Desta forma, enquanto este último MP está processando, tem-se início o preenchimento da memória local para o próximo processamento.

### 3.3. Comparador

O comparador do núcleo é capaz de realizar a comparação dos resultados de SAD dos 3 MPs paralelamente e dar como resultado o menor valor de SAD e seu respectivo vetor

de movimento. Este componente, detalhado na Figura 6, opera em um *pipeline* de 3 estágios. Dessa forma, leva-se 3 ciclos para obter o primeiro resultado e após cada ciclo, uma nova comparação já foi efetuada. Além dos resultados de SAD, são passados como entrada para o componente as coordenadas (x, y) do casamento atual, para que seja possível ser identificado o vetor de movimento relativo ao melhor casamento.

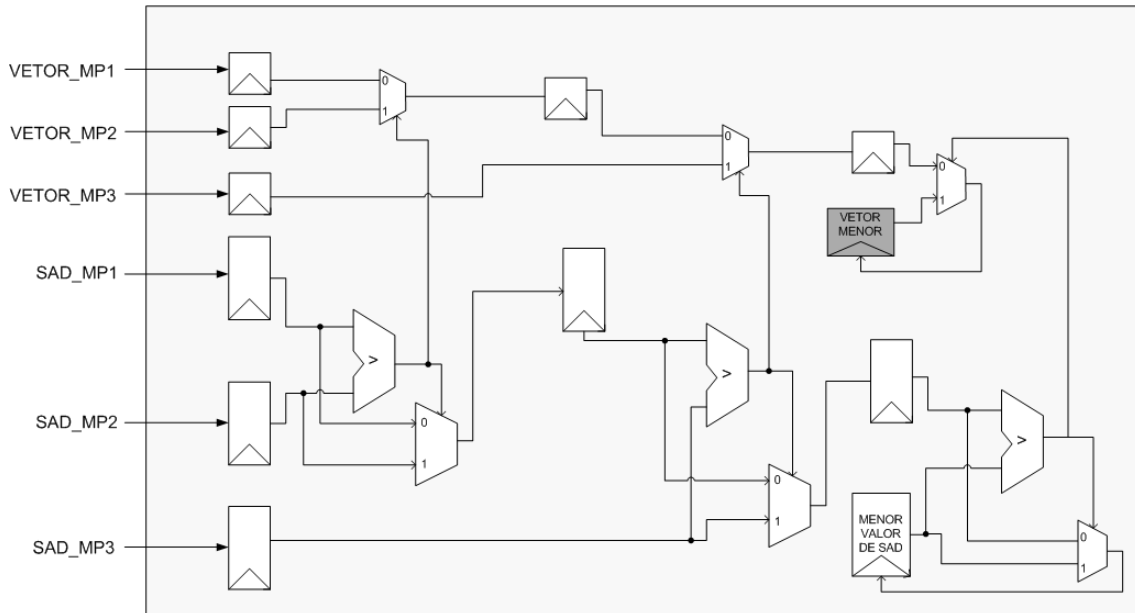


Figura 6. Detalhamento arquitetural do componente Comparador

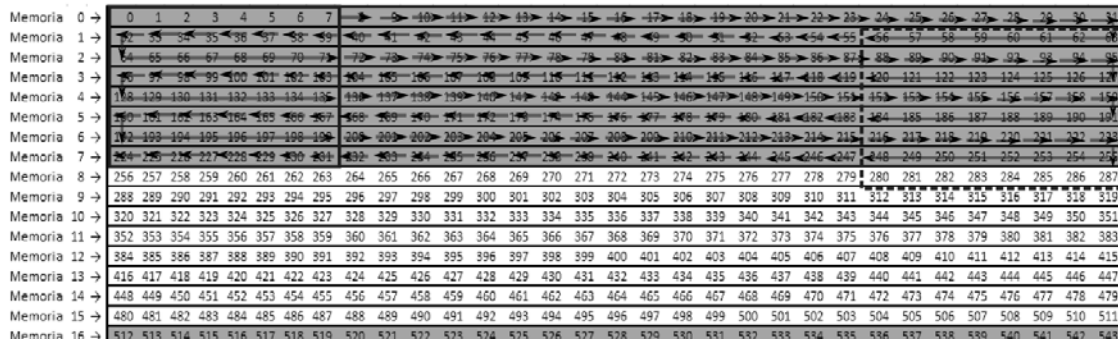
### 3.4 Memórias

São 32 módulos de memória que constituem o núcleo. Cada um é responsável por armazenar 1/32 (uma linha) da área de pesquisa 32x32. As memórias foram idealizadas para funcionarem com o mecanismo de *dual-port*. Na escrita de dados, cada memória é interpretada como tendo 8 palavras de 32 bits. Já na leitura de dados, cada memória foi definida como possuindo 32 palavras de 8 bits (cada palavra representando um pixel).

Adotando esta forma, torna-se possível que a escrita nas memórias locais seja compatível com a leitura que deve ser feita na memória externa à arquitetura, onde estão armazenados os quadros da imagem. Nesta memória externa, a leitura do quadro deve ser realizada da esquerda para a direita e de cima para baixo. Enquanto torna a escrita compatível, este procedimento permite a varredura em zig-zag idealizada para este núcleo, que realiza um reaproveitamento dos dados de um casamento atual para a realização do próximo casamento. Como citado anteriormente, na Busca Completa a diferença de pixels de um casamento para outro é de apenas uma linha/coluna.

A varredura em zig-zag está apresentada na Figura 7. Esta figura apresenta parte da memória local inicializada com um valor qualquer. O quadrado 8x8 representa os dados que se encontram no banco de registradores de referência do MP. Estes dados são referentes ao primeiro casamento da Busca Completa. A primeira seta da linha da Memória 0 representa o deslocamento que deve ser realizado para que seja efetuado o segundo casamento da busca. Neste caso, para a realização do segundo casamento, o seletor do banco de registradores de referência sinaliza que os dados devem sofrer um deslocamento para a esquerda e novos dados lidos da memória são gravados no banco.

As setas seguintes representadas na figura são referentes aos demais casamentos/deslocamentos a serem realizados.



**Figura 7. Varredura da área de pesquisa em zig-zag**

A cada linha, são realizados 25 casamentos entre o bloco atual e o bloco de referência. Ao chegar ao 25º casamento, é feito um deslocamento para baixo no banco de registradores de referência, contando mais um casamento, e mais 24 casamentos são realizados para a segunda linha. Na segunda linha, os deslocamentos são realizados para a esquerda. Este procedimento é realizado até finalizar os casamentos destinados a cada MP.

Esta figura apresenta os casamentos a serem realizados pela MPI. O quadrado tracejado na imagem representa os dados que devem estar no banco de registradores de referência quando a MPI estiver realizando o 26º casamento, ou ainda, o 1o casamento da segunda linha da área de pesquisa.

#### 4. Avaliação de Desempenho da Arquitetura

Uma vez feito o primeiro preenchimento das memórias e do banco de registradores de bloco, a arquitetura é capaz de processar 1 vetor de movimento a cada 241 ciclos. Operando a uma frequência máxima 180 MHz, são processados em média 51 quadros por segundo de uma imagem HDTV (1280x720). Com este resultado percebe-se que a arquitetura, tal como ela foi proposta, é capaz de processar imagens HDTV (1280x720) em tempo real com uma folga considerável. A Tabela 1 mostra a taxa de processamento para outros tamanhos de imagem, bem como a frequência necessária para processamento em tempo real (30 quadros/s).

**Tabela 1. Taxa de processamento de quadros por resolução de imagem**

Resolução	Quadros/s	Frequência mínima para tempo real (MHz)
QCIF (176x144)	1886,08	2,86
QVGA (320x240)	622,40	8,67
SD (720x480)	138,31	39,04
720p (1280x720)	51,86	104,11
1080p HD(1920x1080)	23,05	234,25

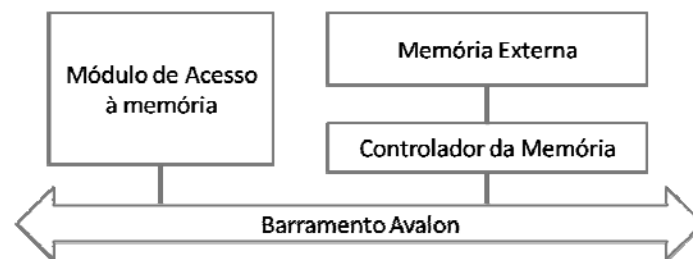
Como os dados apresentam, ainda não é possível atingir taxa de processamento para 1080p. Dessa forma, ainda é preciso reavaliar a arquitetura de modo a cogitar a questão do aumento de área para melhorar o desempenho obtido.

Embora o núcleo seja capaz de processar imagens 720p HDTV em tempo real, um fator muito importante foi desconsiderado para o cálculo deste tempo: os acessos à memória externa. Alguns trabalhos relacionados apresentam suas arquiteturas para estimação de movimento, entretanto desconsideram este gargalo de acesso. Para fazer a verificação deste impacto no desempenho foi feito um estudo referente ao tempo necessário para o preenchimento das memórias locais que constituem a arquitetura apresentada neste trabalho. Este estudo será apresentado na próxima seção.

#### 4. Estudo sobre o impacto do acesso à memória externa

Para este estudo foi utilizado o kit de prototipagem Altera DE2 que possui um FPGA da família Cyclone II (EP2C35F672C6) e conta com duas memórias externas disponíveis: uma memória SRAM de 512 Kb e uma SDRAM de 8 Mb. Estas memórias possuem palavras de 16 bits. Para avaliar o custo de comunicação com as duas memórias externas foi desenvolvido um sistema composto por um Módulo de Acesso à memória, e um controlador de memória, ligado à memória externa, como apresentado na Figura 8.

Este sistema foi integrado utilizando o SOPC Builder [Altera Corporation 2009], que possibilita a fácil integração de módulos para criar sistemas em chip rapidamente. Como mecanismo de interconexão nesta ferramenta é utilizado o barramento Avalon. Os controladores utilizados para as memórias SRAM e SDRAM são módulos disponibilizados pela Altera para a utilização com seus dispositivos compatíveis.



**Figura 8. Sistema para análise do custo de comunicação com a memória externa**

O Módulo de Acesso à Memória realiza as requisições necessárias para o preenchimento das memórias locais que constituem a arquitetura apresentada na seção 3. Para medir o tempo de acesso foi inserido um contador neste módulo que computa o tempo gasto para realizar as leituras na memória e obter o custo de realização desta tarefa. A Tabela 2 apresenta os resultados obtidos. A

Tabela 3 re-apresenta algumas das informações da Tabela 1, considerando a frequência de 180Mhz e incluindo o tempo gasto com acesso à memória externa.

**Tabela 2. Tempo em ciclos para preenchimento das memórias locais**

Memória Externa	Tempo em ciclos
SRAM	2044
SDRAM	8701

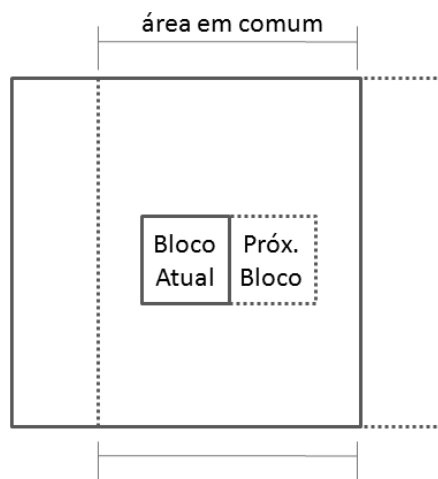


**Tabela 3. Taxa de processamento de quadros com acesso à memória externa**

Resolução	Quadros/s SRAM	Quadros/s SDRAM
QCIF (176x144)	165,77	42,36
SD (720x480)	12,15	3,01
720p (1280x720)	4,55	1,16
1080p HD(1920x1080)	3,03	0,77

Pela tabela apresentada é possível verificar que o acesso à memória externa não é um fator a ser simplesmente desconsiderado, quando se trata de um módulo de estimação de movimento. A estimação de movimento é conhecida por sua grande complexidade computacional, que necessita realizar uma grande quantidade de cálculos para retornar o vetor de movimento. No estudo de caso apresentado, esta quantidade de cálculos chega a ser apenas 11,7% do custo da comunicação com a memória externa SRAM e pouco mais de 2% no caso da SDRAM. Vale salientar que o estudo de caso foi realizado no kit de prototipagem disponível no grupo e se trata de um equipamento lento.

Como estratégia para minimizar a necessidade de acessos à memória externa, esta arquitetura utilizará o reaproveitamento dos dados já existentes na memória local quando da realização da busca de um bloco em sua respectiva área de pesquisa. Na realização da busca do próximo bloco 8x8 em uma região 32x32, 75% dos dados que serão utilizados no próximo processamento já se encontram disponíveis na memória local. Isto porque um quadro é dividido em blocos não sobrepostos, mas as áreas de pesquisa de dois blocos possuem dados em comum. Este esquema está representado na Figura 9. Apenas no caso em que o bloco se encontra na extremidade do quadro é que não é possível realizar reaproveitamento para o próximo processamento.

**Figura 9. Sobreposição de área de pesquisa de blocos vizinhos**

## 5. Conclusões e Trabalhos Futuros

Este artigo apresentou uma proposta de arquitetura para um núcleo de estimação de movimento segundo o padrão H.264. Este núcleo utiliza o algoritmo da Busca Completa

para varredura da área de pesquisa, tamanho de bloco 8x8 e área de pesquisa 32x32. A arquitetura é formada por 3 módulos de processamento (MPs), 1 banco de registradores de bloco, um comparador e memória local.

Foi apresentando também um estudo referente ao impacto do acesso à memória externa no desempenho da arquitetura. Para este estudo foi utilizado o *kit* de prototipagem Altera DE2. Dos experimentos concluiu-se que novas estratégias de acesso a memória devem ser utilizadas de modo a permitir que o poder de processamento da arquitetura seja bem utilizado. Comprova-se também que, muitos dos artigos que advogam o uso de algoritmos subótimos, como o da Busca em Diamante (*Diamond Search*) desconsideram o tempo de acesso a memórias externas para aquisição dos dados. Como estes algoritmos realizam um número inferior de cálculos para encontrar um casamento subótimo, o problema da espera dos dados tende a se agravar.

Trabalhos futuros incluem a utilização de um *kit* Xilinx (XUP V2P), para o qual está disponível um controlador de acesso a memória com melhor desempenho [Bonatto, Soares e Susin 2008]. Outro estudo importante a ser realizado é referente ao reuso de dados em comum entre áreas de pesquisa de blocos vizinhos e criação de uma hierarquia interna de memória na arquitetura, com o objetivo de reduzir os custos de acessos a dados. O aumento do paralelismo da arquitetura do ME, com a utilização de cinco módulos de processamento também é uma alternativa a ser estudada. Com este paralelismo, cada MP seria responsável pela execução 125 casamentos, possibilitando processamento em tempo real de imagens 1080p. Através desses estudos será possível analisar a possibilidade de redução de quantidade de acessos à memória externa e por consequência, aumento no desempenho global do módulo de estimação de movimento.

## Referências

- Agostini, L. V. (2007). “Desenvolvimento de Arquiteturas de Alto Desempenho Dedicadas à Compressão de Vídeo Segundo o Padrão H.264/AVC”. 172f. Programa de Pós Graduação em Computação, UFRGS, Porto Alegre.
- Altera Corporation, (2009). “Introduction to SOPC Builder”, [http://www.altera.com/literature/hb/qts/qts\\_qii54001.pdf](http://www.altera.com/literature/hb/qts/qts_qii54001.pdf), November.
- Bhaskaran, V. and Konstantinides, K. (1997). “Image and Video Compression Standars: Algorithms and Architectures”. 2<sup>nd</sup> edition. Kluwer Academic Publishers: Boston.
- Bonatto, A. C., Soares, A. B., Susin, A. A. (2008). “DDR-SDRAM Memory Controller Validation for FPGA Synthesis”. In: 9th IEEE Latin-American Test Workshop, 2008, Puebla. 9th IEEE Latin-American Test Workshop. p. 177-182.
- Deng, L., Gao. W, Hu, M. Z. and Ji, Z. Z. (2005) “An efficient hardware implementation for motion estimation of AVC standard”, Consumer Electronics, IEEE Transactions on , vol.51, no.4, pages 1360-1366.
- Ghanbari, M. (2003). “Standard Codecs: Image Compression to Advanced Video Coding (Telecommunications)”. Institution Electrical Engineers.
- ITU-T – International Telecommunication Union. (2007). “ITU-T Recommendation H.264/AVC (05/03): advanced video coding for generic audiovisual services”.

- Rosa, L. Z. P. (2007). “Investigação sobre Algoritmos para a Estimação de Movimento na Compressão de Vídeos Digitais de Alta Definição: Uma Análise Quantitativa”. 155f. Departamento de Informática, Universidade Federal de Pelotas, Pelotas.
- Shi, Y. Q., Sun, H. (1999). “Image and Video Compression for Multimedia Engineering: Fundamental, Algorithms and Standards”. CRC Press.
- Vanne, J., et al. (2006). “A High-Performance Sum of Absolute Difference Implementation for Motion Estimation.” Circuits and Systems for Video Technology, IEEE Transactions on. v. 16, n. 7, p. 876-883.
- Verma, R., Akoglu, A. (2008). “A coarse grained and hybrid reconfigurable architecture with flexible NoC router for variable block size motion estimation”. Parallel and Distributed Processing. IPDPS 2008. IEEE International Symposium, p. 1-8.
- Yi, X., Ling, N. (2005). “Rapid block-matching motion estimation using modified diamond search algorithm”. Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium, p. 5489-5492 Vol. 5486