

ERRATA

SILVA, L. M. de L. **Implementação Física de Arquiteturas de Hardware para a Decodificação de Vídeo Digital Segundo o Padrão H.264/AVC**. 136 f. Dissertação (Mestrado em Ciência da Computação) - Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2010.

Folha	Linha	Onde se lê	Leia-se
19	2	Este utiliza	Este sistema utiliza
19	13	destas arquiteturas	dessas arquiteturas
19	15-16	que juntas formam	que, combinadas, formam
19	16	H.264 chamado	H.264, chamado
19	17	Além destas, também foi fisicamente implementada	Além dessas, também passaram pelo fluxo de implementação física
19	24	do filtro	do filtro redutor redutor de efeito de bloco
19	32	comparado com	quando comparada com
21	4	System) for	System) standard for
21	15	the architectures parser	the parser
21	16	and transforms	and transforms architectures
21	17	called intra-only	called intra-only decoder
21	18	it it also	it was also
21	22	the architecture	the architectures
23	6	A compressão	Assim, a compressão
23	7	contidas neste e que	contidas neles, as quais
23	8	meio físico	meio físico (AGOSTINI, 2007)
23	9-17	Para um vídeo com resolução de 720x480 pixels a 30 quadros por segundo (usado em televisão digital com definição	Nesse contexto, conforme explica Agostini (2007, p. 25), considerando vídeos com

		<p>normal – SDTV e em DVDs), utilizando 24 bits por pixel, a taxa necessária para a transmissão sem compressão seria aproximadamente 249 milhões de bits por segundo (249 Mbps). Para armazenar uma sequência de curta duração, com 10 minutos, seriam necessários quase 19 bilhões de bytes (19GB). Para vídeos com resolução de 1920x1080 pixels a 30 quadros por segundo (usado em televisão digital com alta definição ou HDTV), com 24 bits por pixel, a taxa de transmissão sobe para 1,5 bilhões de bits por segundo (1,5 Gbps) e seriam necessários 112 bilhões de bytes (112 GB) para armazenar um vídeo com 10 minutos de duração.</p>	<p>resolução de 720x480 pixels a 30 quadros por segundo (usado em televisão digital com definição normal – SDTV e em DVDs), utilizando 24 bits por pixel, a taxa necessária para a transmissão sem compressão seria aproximadamente 249 milhões de bits por segundo (249 Mbps). Para armazenar uma sequência de curta duração, com 10 minutos, seriam necessários quase 19 bilhões de bytes (19GB). Para vídeos com resolução de 1920x1080 pixels a 30 quadros por segundo (usado em televisão digital com alta definição ou HDTV), com 24 bits por pixel, a taxa de transmissão sobe para 1,5 bilhões de bits por segundo (1,5 Gbps) e seriam necessários 112 bilhões de bytes (112 GB) para armazenar um vídeo com 10 minutos de duração.</p>
23	21-23	<p>Ele é considerado o estado-da-arte em termos de compressão de vídeo, introduzindo um conjunto de ferramentas inovadoras em relação a padrões anteriores.</p>	<p>Ele é considerado o “estado-da-arte em termos de compressão de vídeo, introduzindo um conjunto de ferramentas inovadoras em relação a padrões anteriores” (DINIZ, 2009, p. 17).</p>
23	25	<p>quatro vezes (WIEGAND, 2003)</p>	<p>quatro vezes (DINIZ, 2009)</p>
23	36	<p>vários módulos/etapas</p>	<p>várias etapas</p>
23	37	<p>Estas etapas são definidas</p>	<p>Essas arquiteturas podem ser definidas</p>
24	10	<p>Esta transição</p>	<p>Essa transição</p>

24	12	um dos objetivos	um de seus objetivos
24	16	âmbito do projeto	âmbito desse projeto
24	19	inversas (AGOSTINI, 2006)	inversas (AGOSTINI et al., 2006)
24	20	bloco (ROSA, 2009)	bloco (ROSA; BAMPI; SUSIN, 2009)
24	21	desenvolvida em (AZEVEDO, 2007)	desenvolvida no trabalho de Azevedo et al. (2007)
24	22	desenvolvida em (ZATT, 2008a)	desenvolvida no trabalho de Zatt (2008a)
24	25	H.264/AVC (AGOSTINI, 2007).	H.264/AVC (AGOSTINI, 2007, p. 46).
24	27	trabalho de (PEREIRA, 2006)	trabalho de Pereira (2006)
24	29	integradas gerando	integradas, gerando
24	31	foram integrados	foram integradas
24	32	<i>intra-only</i> , que é	<i>intra-only</i> .
24	33-34	linguagem VHDL, validadas através de	linguagem VHDL (<i>VHSIC Hardware Description Language</i>) validadas por meio de
24	36	de realizar	de se realizar
25	8	H.264 <i>intra-only</i>	H.264 <i>intra-only</i> (PEREIRA, 2009, p. 26)
25	13	Elas juntas formam	Elas, combinadas, formam
25	14	Alem destas,	Além dessas,
25	15	em ASIC, a	em ASIC a
25	18	literatura, para	literatura para
25	25	analisando assim a	analisando, assim, a
25	36	trabalho. Alguns	trabalho e alguns
27	3	vídeo e dá	vídeo, e dá

27	4	Aqui é apresentado	Nele é apresentado
27	10	detalhada deste	detalhada dele
27	11	de Agostini (AGOSTINI, 2007), assim como em (SULLIVAN, 2005; RICHARDSON, 2003).	de Agostini (2007), assim como nos trabalhos de Sullivan e Wiegand (2005) e Richardson (2003)
27	14-16	Um sistema para representar cores é chamado de espaço de cores, e a definição do espaço de cores a ser utilizado para representar um vídeo é essencial para a eficiência da codificação deste vídeo.	Segundo Agostini (2007, p. 29), “um sistema para representar cores é chamado de espaço de cores, e a definição do espaço de cores a ser utilizado para representar um vídeo é essencial para a eficiência da codificação deste vídeo.”
27	16-17	São vários os espaços de cores usados para representar imagens digitais, tais como: RGB, HSI e YCbCr (SHI, 1999).	Nesse contexto, “são vários os espaços de cores usados para representar imagens digitais, tais como: RGB, HSI e YCbCr (SHI, 1999 apud AGOSTINI, 2007, p. 29).”
27	17-21	O RGB (<i>red, green e blue</i>) representa, em três matrizes distintas, as três cores primárias captadas pelo sistema visual humano: vermelho, verde e azul. No espaço de cores YCbCr, as três componentes utilizadas são luminância (Y), que define a intensidade luminosa ou o brilho; croma azul (Cb) e croma vermelho (Cr) (BHASKARAN, 1997).	O RGB (<i>red, green e blue</i>), um dos mais relevantes deles, representa, em três matrizes distintas, as três cores primárias captadas pelo sistema visual humano: vermelho, verde e azul. No espaço de cores YCbCr, as três componentes utilizadas são luminância (Y), que define a intensidade luminosa ou o brilho; croma azul (Cb) e croma vermelho (Cr) (BHASKARAN, 1997 apud AGOSTINI, 2007, p. 29).
27	22-28	Os componentes R, G e B possuem um elevado grau de	Sobre esse espaço de cores, explica Agostini (2007, p. 30): os componentes R, G e B

		<p>correlação, o que não é desejável do ponto de vista da compressão de vídeos. Por isso, a compressão é aplicada para espaços de cores do tipo luminância e croma, como o YCbCr (RICHARDSON, 2002). Outra vantagem do espaço de cor YCbCr sobre o espaço RGB é que, no espaço YCbCr, a informação de cor está completamente separada da informação de brilho. Deste modo, estas informações podem ser tratadas de forma diferenciada pelos codificadores de imagens estáticas e de vídeos (AGOSTINI, 2007).</p>	<p>possuem um elevado grau de correlação, o que não é desejável do ponto de vista da compressão de vídeos. Por isso, a compressão é aplicada para espaços de cores do tipo luminância e croma, como o YCbCr (RICHARDSON, 2002). Outra vantagem do espaço de cor YCbCr sobre o espaço RGB é que, no espaço YCbCr, a informação de cor está completamente separada da informação de brilho. Deste modo, estas informações podem ser tratadas de forma diferenciada pelos codificadores de imagens estáticas e de vídeos.</p>
27	30	Estas técnicas	Essas técnicas
27	33	Nesta última,	Nessa última,
28	23-36	<p>Assim, existem 3 tipos de subamostragem: 4:4:4, o 4:2:2 e o 4:2:0.</p> <p>No formato 4:4:4, para cada quatro amostras de luminância (Y), existem quatro amostras de croma azul (Cb) e quatro amostras de croma vermelha (Cr). Por isso, os três componentes de cor possuem a mesma resolução, existe uma amostra de cada elemento de cor para cada pixel da imagem e, assim, a subamostragem não é aplicada. No formato 4:2:2, para cada quatro amostras de Y</p>	<p>Assim, existem três tipos de subamostragem: a 4:4:4, o 4:2:2 e o 4:2:0. Dessa forma,</p> <p>no formato 4:4:4, para cada quatro amostras de luminância (Y), existem quatro amostras de croma azul (Cb) e quatro amostras de croma vermelha (Cr). Por isso, os três componentes de cor possuem a mesma resolução, existe uma amostra de cada elemento de cor para cada pixel da imagem e, assim, a subamostragem não é aplicada. No formato 4:2:2, para cada quatro amostras de Y na direção</p>

		<p>na direção horizontal, existem apenas duas amostras de Cb e duas amostras de Cr. Neste caso, as amostras de crominância possuem a mesma resolução vertical das amostras de luminância, mas possuem metade da resolução horizontal. No formato 4:2:0, para cada quatro amostras de Y, existe apenas uma amostra de Cb e uma amostra de Cr. Neste caso, as amostras de crominância possuem metade da resolução horizontal e metade da resolução vertical, com relação às amostras de luminância (AGOSTINI, 2007);</p>	<p>horizontal, existem apenas duas amostras de Cb e duas amostras de Cr. Neste caso, as amostras de crominância possuem a mesma resolução vertical das amostras de luminância, mas possuem metade da resolução horizontal. No formato 4:2:0, para cada quatro amostras de Y, existe apenas uma amostra de Cb e uma amostra de Cr. Neste caso, as amostras de crominância possuem metade da resolução horizontal e metade da resolução vertical do que as amostras de luminância (AGOSTINI, 2007, p. 30);</p>
28-29	37-2	<p>Transformadas e Quantização - Os olhos humanos têm maior perceptividade para mudanças suaves em vídeos do que para texturas detalhadas. O termo frequência espacial determina a taxa com que os pixels mudam de um lugar para outro. Neste contexto, altas frequências espaciais significam uma grande diferença entre as posições vizinhas de pixels, ao passo que baixa frequência espacial significa modificações mais suaves. A aplicação de transformadas sobre uma determinada área é uma operação sem perdas de informação que recebe como entrada uma matriz de pixels e produz a frequência espacial</p>	<p>Transformadas e Quantização -</p> <p>Os olhos humanos têm maior perceptividade para mudanças suaves em vídeos do que para texturas detalhadas. O termo frequência espacial determina a taxa com que os pixels mudam de um lugar para outro. Neste contexto, altas frequências espaciais significam uma grande diferença entre as posições vizinhas de pixels, ao passo que baixa frequência espacial significa modificações mais suaves. A aplicação de transformadas sobre uma determinada área é uma operação sem perdas de informação que recebe como entrada uma matriz de pixels e produz a frequência espacial dessa matriz. A aplicação da quantização é uma operação com perdas que</p>

		dessa matriz. A aplicação da quantização é uma operação com perdas que descarta as partes da imagem que são menos perceptíveis aos olhos. (PEREIRA, 2009).	descarta as partes da imagem que são menos perceptíveis aos olhos. (PEREIRA, 2009, p. 11, tradução nossa).
29	8	qual a combinação	qual combinação
29	21	H.264/AVC (AGOSTINI, 2007)	H.264/AVC (AGOSTINI, 2007, p. 45)
29	34	decodificação, para	decodificação para
30	4-5	(Filtro) do processo de decodificação também	(Filtro), do processo de decodificação, também
30	5-7	A função deles é suavizar o efeito de bloco do quadro reconstruído antes de ele ser usado para fazer a predição de um novo macrobloco, utilizando predição do tipo inter-quadros.	A função dele, segundo Staehler (2006, p. 92), “é suavizar o efeito de bloco do quadro reconstruído antes de ele ser usado para fazer a predição de um novo macrobloco do tipo inter”.
30	9-25	Em 1990, o ITU-T (<i>International Telecommunication Union - Telecommunication</i>) (ITU-T, 2010a) publicou o padrão H.261 (ITU-T, 1990), estabelecendo assim as ferramentas básicas para os CODECS (codificador e decodificador) atuais. Depois do H.261, outros padrões foram publicados nos anos seguintes: O MPEG (<i>Motion Picture Experts Group</i>) da ISO (<i>International Organization for Standardization</i>) (ISO/IEC, 2010) publicou o MPEG-1	Conforme elucidada Pereira (2009, p. 13, tradução nossa), em 1990, o ITU-T (<i>International Telecommunication Union - Telecommunication</i>) (ITU-T, 2010a) publicou o padrão H.261 (ITU-T, 1990), estabelecendo, assim, as ferramentas básicas para os CODECS (codificador e decodificador) atuais. Depois do H.261, outros padrões foram publicados nos anos seguintes: O MPEG (<i>Motion Picture Experts Group</i>) da ISO (<i>International Organization for Standardization</i>)

		<p>(ISO/IEC, 1993) e depois o MPEG-2, que foi padronizado como H.262 (ITU-T, 1994). O H.262/MPEG-2 obteve grande sucesso e se tornou mais popular que outros padrões como ITU-T H.263 e ISO MPEG4 Parte 2.</p> <p>Em 2001, o ITU-T e ISO se juntaram e criaram o JVT (<i>Joint Video Team</i>) (ITU-T, 2010b) com o intuito de elaborar um novo CODEC de alto desempenho, aprovado em 2003 (ITU-T, 2003) com uma extensão 2005 (ITU-T, 2005). Este CODEC é chamado de MPEG4 Parte 10 (nome dado pelo ISO) e H.264 (nome dado pelo ITU-T). Este padrão possui ferramentas/técnicas que proporcionam um desempenho consideravelmente maior que os padrões anteriores, chegando a duplicar a taxa de compressão em relação ao padrão mais eficiente até então, que era o MPEG-2. Softwares de referência também estão disponíveis para testar o padrão (PEREIRA, 2009).</p>	<p>(ISO/IEC, 2010) publicou o MPEG-1 (ISO/IEC, 1993) e depois o MPEG-2, que foi padronizado como H.262 (ITU-T, 1994). O H.262/MPEG-2 obteve grande sucesso e se tornou mais popular que outros padrões como ITU-T H.263 e ISO MPEG4 Parte 2.</p> <p>Em 2001, o ITU-T e ISO se juntaram e criaram o JVT (<i>Joint Video Team</i>) (ITU-T, 2010b) com o intuito de elaborar um novo CODEC de alto desempenho, aprovado em 2003 (ITU-T, 2003) com uma extensão 2005 (ITU-T, 2005). Este CODEC é chamado de MPEG4 Parte 10 (nome dado pelo ISO) e H.264 (nome dado pelo ITU-T). Este padrão possui ferramentas/técnicas que proporcionam um desempenho consideravelmente maior que os padrões anteriores, chegando a duplicar a taxa de compressão em relação ao padrão mais eficiente até então, que era o MPEG-2. Softwares de referência também estão disponíveis para testar o padrão.</p>
30	28-30	<p>Perfis estão relacionados ao conjunto de características que um CODEC utiliza para codificar um vídeo, enquanto níveis estão relacionados aos requisitos de processamento e memória.</p>	<p>Segundo Pereira (2009, p. 15, tradução nossa), “perfis estão relacionados ao conjunto de características que um CODEC utiliza para codificar um vídeo, enquanto níveis estão relacionados aos requisitos de processamento e memória.”</p>

30	34	campos, um	campos: um
31	1	três tipos:	três tipos, conforme explica Pereira (2009):
31	4-8	Macrobloco tipo P (preditivo) - utiliza predição inter-quadros por meio de quadros de referência previamente codificados e armazenados na lista 0. Este tipo de macrobloco pode ser dividido em partições de tamanhos 16x16, 16x8, 8x16 e 8x8. Este último pode ser também subdividido em partições de sub-macrobloco de 8x4, 4x8 e 4x4.	Macrobloco tipo P (preditivo) - utiliza predição inter-quadros por meio de quadros de referência previamente codificados e armazenados na lista 0. Este tipo de macrobloco pode ser dividido em partições de tamanhos 16x16, 16x8, 8x16 e 8x8. Este último pode ser também subdividido em partições de sub-macrobloco de 8x4, 4x8 e 4x4 (PEREIRA, 2009, p. 14, tradução nossa).
31	13-16	Em um quadro, os macroblocos são agrupados em um ou mais <i>slices</i> , que podem ser do tipo I, P, B, SP (<i>Switching P</i>) ou SI (<i>Switching I</i>). <i>Slices</i> do tipo I podem somente ter macroblocos do tipo I, <i>slices</i> do tipo P podem ter macroblocos do tipo I e P, e <i>slices</i> do tipo B podem ser os três tipo de macroblocos, I, P e B.	Nesse contexto, conforme exposto em Pereira (2009, p. 14, tradução nossa), em um quadro, os macroblocos são agrupados em um ou mais <i>slices</i> , que podem ser do tipo I, P, B, SP (<i>Switching P</i>) ou SI (<i>Switching I</i>). <i>Slices</i> do tipo I podem somente ter macroblocos do tipo I, <i>slices</i> do tipo P podem ter macroblocos do tipo I e P, e <i>slices</i> do tipo B podem ser os três tipo de macroblocos, I, P e B.
31	21	<i>slice</i> (PORTO, 2008).	<i>slice</i> (PORTO, 2008, p. 24).
31	26-27	A FRExt produziu um grupo de quatro novos perfis chamados coletivamente de perfis <i>High</i> (SULLIVAN, 2004).	Assim, “o FRExt produziu um grupo de quatro novos perfis chamados coletivamente de perfis <i>High</i> ” (SULLIVAN et al, 2004, apud DINIZ, 2009, p. 31).
31	33	este suporta	ele suporta
31	33	foram um conjunto	formam um conjunto

31	36-37	vídeo (SULLIVAN, 2004).	vídeo (SULLIVAN et al., 2004).
32	1-7	O perfil <i>High</i> (HP) inclui vídeos com 8 bits por amostra e com relação de cores 4:2:0. O perfil <i>High</i> 10 (Hi10P) suporta vídeos com 10 bits por amostra, também com uma relação de cores 4:2:0. O perfil <i>High</i> 4:2:2 (H422P) inclui suporte à relação de cor 4:2:2 e vídeos a 10 bits por amostra. Finalmente, o perfil <i>High</i> 4:4:4 (H444P) dá suporte à relação de cores 4:4:4 (ou seja, sem nenhuma subamostragem), suporte a vídeos com até 12 bits por amostra e, adicionalmente, suporta a codificação sem perdas em determinadas regiões do vídeo (AGOSTINI, 2007).	Sobre esses perfis, explica Agostini (2007, p. 41): o perfil <i>High</i> (HP) inclui vídeos com 8 bits por amostra e com relação de cores 4:2:0. O perfil <i>High</i> 10 (Hi10P) suporta vídeos com 10 bits por amostra, também com uma relação de cores 4:2:0. O perfil <i>High</i> 4:2:2 (H422P) inclui suporte à relação de cor 4:2:2 e vídeos a 10 bits por amostra. Finalmente, o perfil <i>High</i> 4:4:4 (H444P) dá suporte à relação de cores 4:4:4 (ou seja, sem nenhuma subamostragem), suporte a vídeos com até 12 bits por amostra e, adicionalmente, suporta a codificação sem perdas em determinadas regiões do vídeo.
32	11	H.264/AVC (AGOSTINI, 2007)	H.264/AVC (AGOSTINI, 2007, p. 41)
32	14	H.264/AVC (ZATT, 2008).	H.264/AVC (ZATT, 2008a, p. 28).
32-33	16-2	Além da divisão em diversos perfis, o padrão H.264/AVC também define 16 diferentes níveis em função da taxa de processamento e da quantidade de memória necessária para cada implementação. Com a definição do nível utilizado, é possível deduzir o número máximo de quadros de referência e a máxima taxa de bits que pode ser utilizada (SULLIVAN, 2004).	Ainda nesse contexto, segundo Sullivan et al.(2004, apud ZATT, 2008a, p. 27) Além da divisão em diversos perfis, o padrão H.264/AVC também define 16 diferentes níveis em função da taxa de processamento e da quantidade de memória necessária para cada implementação. Com a definição do nível utilizado, é possível deduzir o número máximo de quadros de referência e a máxima taxa de bits que

			pode ser utilizada.
33	12	H.264/AVC (AGOSTINI, 2007)	H.264/AVC (AGOSTINI, 2007, p. 46)
33	28	em (AGOSTINI, 2007)	no trabalho de doutorado de Agostini (2007)
33	34	Golomb) (SILVA, 2007)	Golomb) (SILVA et al., 2007)
34	5	é utilizado	é utilizada
34	6-7	Este último não é suportado	Esta última não é suportada
34	7-8	o CAVLC	a CAVLC
34	14	em (DEPRÁ, 2009; AGOSTINI, 2007).	nos trabalhos de Deprá (2009) e Agostini (2007).
34	19-24	Ela realiza a correção de escala nas informações que passaram pelas transformadas diretas na codificação. Como na etapa de quantização na codificação ocorrem perdas, os resíduos obtidos após esta etapa de quantização inversa não são os mesmos que eram antes da etapa de quantização no codificador. Em cada bloco 4x4, a amostra superior esquerda é chamada de elemento DC. Os outros elementos são chamados de elementos AC.	Conforme lição de Pereira (2009, p. 19, tradução nossa), ele realiza a correção de escala nas informações que passaram pelas transformadas diretas na codificação. Como na etapa de quantização na codificação ocorrem perdas, os resíduos obtidos após esta etapa de quantização inversa não são os mesmos que eram antes da etapa de quantização no codificador. Em cada bloco 4x4, a amostra superior esquerda é chamada de elemento DC. Os outros elementos são chamados de elementos AC.
34	28-31	O algoritmo de quantização inversa (Q^{-1}) consiste na multiplicação das entradas por uma constante, a adição dos resultados por outra constante e	De acordo com Pereira (2009, p. 19, tradução nossa), o algoritmo de quantização inversa (Q^{-1}) consiste na multiplicação das entradas

		finalmente o deslocamento da adição por outra constante. Estas constantes são influenciadas diretamente pelo parâmetro de quantização (QP)	por uma constante, a adição dos resultados por outra constante e finalmente o deslocamento da adição por outra constante. Estas constantes são influenciadas diretamente pelo parâmetro de quantização (QP)
34	31	que é	O QP é
34	32	na codificação	na codificação (AGOSTINI, 2007)
34	35-39	O QP pode variar de 0 a 51, e para cada QP, existe um Qstep. Os primeiros seis valores de Qstep, relativos aos seis primeiros QP, são definidos pelo padrão, como está apresentado na Tabela 3.4. Os demais Qsteps podem ser derivados dos seis primeiros, pois o Qstep dobra de valor a cada variação de 6 no valor do QP. Então o $Qstep_{(6)}$ é igual $Qstep_{(0)} \times 2$.	Conforme exposto em Agostini (2007, p. 58), o QP pode variar de 0 a 51, e para cada QP, existe um Qstep. Os primeiros seis valores de Qstep, relativos aos seis primeiros QP, são definidos pelo padrão, como está apresentado na Tabela 2.1. Os demais Qsteps podem ser derivados dos seis primeiros, pois o Qstep dobra de valor a cada variação de 6 no valor do QP. Então o $Qstep_{(6)}$ é igual $Qstep_{(0)} \times 2$.
34	42	Fonte: AGOSTINI, 2007. p. 58.	Fonte: AGOSTINI (2007, p. 58).
35	6-12	As operações das transformadas inversas são muito semelhantes às operações das transformadas diretas. Transformadas chamadas de Hadamard 2x2 e Hadamard 4x4 são calculadas diretamente sobre os coeficientes DC provenientes da etapa de quantização, antes	Sobre as operações das transformadas inversas, explica Diniz (2009, p. 41): são muito semelhantes às operações das transformadas diretas. Transformadas chamadas de Hadamard 2x2 e Hadamard 4x4 são calculadas diretamente sobre os coeficientes DC provenientes da etapa de quantização, antes da

		da etapa de quantização inversa. Depois destas operações, os coeficientes são entregues à etapa de quantização inversa para só então serem processados pela etapa de DCT 2-D inversa (IDCT 2-D) (DINIZ, 2009).	etapa de quantização inversa. Depois destas operações, os coeficientes são entregues à etapa de quantização inversa para só então serem processados pela etapa de DCT 2-D inversa (IDCT 2-D).
35	15-20	O módulo de predição intra-quadro é responsável pela predição de um quadro de saída baseado em valores previamente codificados do <i>slice</i> atual dos pixels acima e à esquerda do bloco a ser codificado. Este tipo de predição é utilizado em macroblocos do tipo I. Para amostras de luminância, o tamanho do bloco para predição pode ser de 16x16 ou 4x4. Há 9 diferentes modos de predição de blocos 4x4 são utilizados e 4 modos quando blocos 16x16 são utilizados.	Conforme extraído de Richardson (2003, apud AGOSTINI, 2007, p. 54): o módulo de predição intra-quadro do padrão H.264/AVC é responsável por realizar a predição nos macroblocos do tipo I. Esta predição é baseada nos valores previamente codificados do <i>slice</i> atual dos pixels acima e à esquerda do bloco a ser codificado. A predição intra-quadro para amostras de luminância pode ser utilizada sobre blocos 4x4 ou 16x16. Existem nove diferentes modos de predição intra-quadro para blocos 4x4 e quatro modos para a predição sobre blocos 16x16.
35	21-26	A Figura 2.6 apresenta os nove tipos diferentes de codificação no modo intra-quadro para blocos 4x4 de luminância. Os modos 0 e 1 fazem uma simples extrapolação (uma cópia) dos pixels das bordas verticais ou horizontais para todas as posições do bloco. O modo DC (2) faz uma média entre as amostras das bordas e copia o resultado para todas as	Sobre esses modos, a Figura 2.6 apresenta os nove tipos diferentes de codificação no modo intra-quadro para blocos 4x4 de luminância. Os modos 0 e 1 fazem uma simples extrapolação (uma cópia) dos pixels das bordas verticais ou horizontais para todas as posições do bloco. O modo DC (2) faz uma média entre as amostras das bordas e copia o resultado para todas as posições do bloco.

		posições do bloco. Os demais modos (3 a 8) fazem uma média ponderada das amostras das bordas, de acordo com a direção da seta na Figura 2.6.	Os demais modos (3 a 8) fazem uma média ponderada das amostras das bordas, de acordo com a direção da seta na Figura 2.6 (AGOSTINI, 2007, p. 54).
35	27-29	A predição da crominância é realizada diretamente sobre blocos 8x8 e utiliza 4 modos distintos de predição, mas os dois componentes de crominância utilizam sempre o mesmo modo (AGOSTINI, 2007).	Nesse contexto, segundo Richardson (2003, apud AGOSTINI, 2007, p. 55), “a predição da crominância é realizada diretamente sobre blocos 8x8 e utiliza 4 modos distintos de predição, mas os dois componentes de crominância utilizam sempre o mesmo modo.”
35	31-32	4x4 (AGOSTINI, 2007)	4x4 (AGOSTINI, 2007, p. 55)
36	3	e tipo	e o tipo
36	9	Ambos juntos	Ambos, combinados,
36	11	é parte	é a parte
36	17	a estas	a essas
36	28	referência (AGOSTINI, 2007)	referência (AGOSTINI, 2007, p. 51)
36	31	Em (AZEVEDO, 2007)	No trabalho de Azevedo et al. (2007)
36	32-33	Em (ZATT, 2008a; ZATT, 2008b)	Nos trabalhos de Zatt (2008a) e Zatt et al. (2008b),
36	37	MoCHA (AZEVEDO, 2007)	MoCHA (AZEVEDO et al., 2007)
37	7	em (AZEVEDO, 2007; ZATT, 2008a)	nos trabalhos de Azevedo et al. (2007) e Zatt (2008a)
37	14-15	HP422-MoCHA (ZATT, 2008a)	HP422-MoCHA (ZATT, 2008a, p. 58)
37	17-21	A compensação de movimento reconstrói o quadro atual utilizando como referências as regiões de quadros previamente	De acordo com Agostini (2007, p. 145), a compensação de movimento reconstrói o quadro atual utilizando

		decodificados. As regiões de referência são indicadas por vetores de movimento (<i>Motion Vector – MV</i>). Estes MVs são calculados por meio de um processo chamado de predição de vetores de movimento (<i>Motion Vector Prediction – MVPr</i>).	como referências as regiões de quadros previamente decodificados. As regiões de referência são indicadas por vetores de movimento (<i>Motion Vector – MV</i>). Estes MVs são calculados por meio de um processo chamado de predição de vetores de movimento (<i>Motion Vector Prediction – MVPr</i>).
37	23-24	em (AZEVEDO, 2007)	no trabalho Azevedo et al. (2007)
37	30	seja limitado	seja limitado (ZATT et al., 2010a)
37	34	temporal (ZATT, 2008)	temporal (ZATT et al., 2010a)
38	2	Movimento (ZATT, 2007)	Movimento (ZATT et al., 2010a, p.9)
38	6	Esta largura	Essa largura
38	7	pois quando	pois, quando
38	9	é um	torna-se um
38	16	<i>Count</i>) (ZATT, 2008)	<i>Count</i>) (ZATT, 2008a)
38	19	do MC (ZATT, 2007).	do MC (ZATT, 2008a, p. 72).
39	2	amostras, durante	amostras durante
39	6	HD1080p, em	HD1080p em
39	12	aguardando (ZATT, 2008)	aguardando (ZATT, 2008a)
39	13-14	em (AZEVEDO, 2007; ZATT, 2008a)	nos trabalhos de Azevedo et al. (2007) e Zatt (2008a)
39	18	Fonte: ZATT, 2008a. p. 75.	Fonte: ZATT (2008a, p. 75).
39	21-28	Como imagens são processadas em blocos, segundo o padrão H.264/AVC, é possível que nos blocos reconstruídos, haja	Sobre o Filtro, explica Pereira (2009, p. 23, tradução nossa): como imagens são processadas em blocos,

		efeitos de bloco em suas bordas, devido à forma como estes foram processados. Para reduzir este efeito, um filtro redutor de efeito de bloco é aplicado nas bordas dos blocos no final do processo de decodificação, ou seja, o objetivo do filtro é suavizar o efeito de bloco do quadro reconstruído antes que ele seja utilizado como quadro de referência para fazer a predição de um novo macrobloco do tipo inter-quadros. Este filtro produz um efeito significativo na qualidade subjetiva do vídeo reconstruído (PEREIRA, 2009)	segundo o padrão H.264/AVC, é possível que nos blocos reconstruídos, haja efeitos de bloco em suas bordas, devido à forma como estes foram processados. Para reduzir este efeito, um filtro redutor de efeito de bloco é aplicado nas bordas dos blocos no final do processo de decodificação, ou seja, o objetivo do filtro é suavizar o efeito de bloco do quadro reconstruído antes que ele seja utilizado como quadro de referência para fazer a predição de um novo macrobloco do tipo inter-quadros. Este filtro produz um efeito significativo na qualidade subjetiva do vídeo reconstruído.
39	31	filtro. (ROSA, 2010).	filtro (ROSA; SILVA; BAMPI, 2010, p. 2).
40	2-13	Para cada borda de bloco, o filtro é aplicado aos valores de pixels perpendiculares à borda. A convenção de nomes para os pixels ao redor das bordas é mostrada na Figura 2.12. Tanto os pixels do bloco atualmente sendo processado (bloco Q), quanto os do bloco passado (bloco P), podem sofrer modificações. Os que já tiverem sido modificados por outro estágio de filtragem podem ser modificados novamente em uma operação subsequente. O algoritmo para filtragem é adaptativo, assim, o <i>boundary strength</i> (bS),	Nesse contexto, conforme explicado em Rosa, Silva e Bampi (2010 et al., p. 2, tradução nossa), para cada borda de bloco, o filtro é aplicado aos valores de pixels perpendiculares à borda. A convenção de nomes para os pixels ao redor das bordas é mostrada na Figura 2.12. Tanto os pixels do bloco atualmente sendo processado (bloco Q), quanto os do bloco passado (bloco P), podem sofrer modificações. Os que já tiverem sido modificados por outro estágio de filtragem podem ser modificados novamente em uma operação subsequente. O algoritmo para filtragem é

		<p>parâmetro que define a força de filtragem, é calculado levando em consideração os seguintes elementos: os valores dos pixels, as posições deles dentro do macrobloco, o tipo de predição aplicada (inter ou intra), os vetores de predição (predição inter) e o QP. O bS pode assumir cinco valores diferentes, variando de 0 (sem filtragem) a 4 (filtragem máxima), dependendo do contexto e localização da borda (ROSA, 2010).</p>	<p>adaptativo, assim, o <i>boundary strength</i> (bS), parâmetro que define a força de filtragem, é calculado levando em consideração os seguintes elementos: os valores dos pixels, as posições deles dentro do macrobloco, o tipo de predição aplicada (inter ou intra), os vetores de predição (predição inter) e o QP. O bS pode assumir cinco valores diferentes, variando de 0 (sem filtragem) a 4 (filtragem máxima), dependendo do contexto e localização da borda.</p>
40	15	bordas. (ROSA, 2010).	bordas (ROSA; SILVA; BAMPI, 2010, p.2).
40	18	Em (ROSA, 2010),	No trabalho de Rosa, Bampi e Susin (2009),
41	2	<i>filter</i> (ROSA, 2010).	<i>filter</i> (ROSA; BAMPI; SUSIN, 2009, p. 5).
41	5-12	<p>O módulo <i>transpose</i> é utilizado para converter amostras do bloco alinhadas verticalmente para amostras alinhadas de forma horizontal, permitindo que estas possam ser processadas pelo <i>edge filter</i>. O módulo <i>input buffer</i> é necessário para o reordenamento dos blocos, uma vez que estes são enviados para processamento em uma ordem diferente da que devem ser processados. Os módulos <i>MB buffer</i> e <i>line buffer</i> são utilizados para armazenar blocos e determinadas informações sobre eles (QP, tipo do bloco, <i>offset</i> de</p>	<p>Como detalhado em Rosa, Bampi e Susin (2009, p. 3, tradução nossa),</p> <p>o módulo <i>transpose</i> é utilizado para converter amostras do bloco alinhadas verticalmente para amostras alinhadas de forma horizontal, permitindo que estas possam ser processadas pelo <i>edge filter</i>. O módulo <i>input buffer</i> é necessário para o reordenamento dos blocos, uma vez que estes são enviados para processamento em uma ordem diferente da que devem ser processados. Os módulos <i>MB buffer</i> e <i>line buffer</i> são utilizados para armazenar blocos e determinadas informações sobre eles (QP, tipo do bloco, <i>offset</i> de filtragem,</p>

		filtragem, estado), que não são completamente filtradas.	estado), que não são completamente filtradas.
41	13	os módulos, o trabalho de (ROSA, 2010)	esses módulos, o trabalho de Rosa, Silva e Bampi (2010)
41	16	bloco (ROSA, 2010)	bloco (ROSA; SILVA; BAMPI, 2010, p. 3)
42	1	de (ROSA, 2009)	de Rosa, Bampi e Susin (2009)
42	6	Fonte: ROSA, 2009. p. 9.	Fonte: Rosa, Bampi e Susin (2009, p. 9).
43	7	mestrado de Pereira (PEREIRA, 2009).	mestrado de Pereira (2009).
43	9	<i>intra-only</i> sintetizado em ASIC	<i>intra-only</i> , que passou pelo fluxo de síntese ASIC
43	11-13	Ele suporta decodificação de entropia utilizando algoritmo CAVLC (<i>Context-Based Adaptive Variable Length Coding</i>), algoritmos para quantização inversa, transformadas inversas e predição intra-quadro.	Segundo Pereira (2009, p. 25, tradução nossa), ele “suporta decodificação de entropia utilizando algoritmo CAVLC (<i>Context-Based Adaptive Variable Length Coding</i>), algoritmos para quantização inversa, transformadas inversas e predição intra-quadro.”
43	16	<i>Coding</i>), e	<i>Coding</i>) e
43	22	H.264 <i>intra-only</i>	H.264 <i>intra-only</i> (PEREIRA, 2009, p. 26)
43	24	O <i>bitstream</i>	Conforme exposto em Pereira (2009), o <i>bitstream</i>
44	3	Assim, o <i>bitstream</i>	Assim, o <i>bitstream</i> , de acordo com Pereira (2009),
44	13	para predição	para predição (PEREIRA, 2009)
44	18	Por fim,	Por fim, conforme explicado em Pereira (2009),
44	27	e a Figura 3.2	e, conforme os trabalhos de Bonatto et al. (2010) e Bonatto, Soares e Susin (2010), a

			Figura 3.2
44	30	em FPGA (BONATTO, 2010)	em FPGA (BONATTO et al., 2010)
45	3	Fonte: BONATTO, 2010. p. 4.	Fonte: Bonatto et al. (2010, p. 4).
45	9	por segundo (BONATTO, 2010)	por segundo (BONATTO et al., 2010)
45	20	de entropia	de entropia (PEREIRA, 2009, p. 27)
45	22-26	Ele é organizado como um conjunto de máquinas de estado em que os dados são decodificados dependendo do estado atual. O sub-módulo <i>Parser Control</i> possui a principal máquina de estados, ativando os outros sub-módulos de acordo com o estado atual. Os sub-módulos <i>Slice Header</i> e <i>Slice Data</i> têm suas próprias máquinas de estado para controle das operações internas a eles.	O <i>parser</i> é organizado da seguinte forma: como um conjunto de máquinas de estado em que os dados são decodificados dependendo do estado atual. O sub-módulo <i>Parser Control</i> possui a principal máquina de estados, ativando os outros sub-módulos de acordo com o estado atual. Os sub-módulos <i>Slice Header</i> e <i>Slice Data</i> têm suas próprias máquinas de estado para controle das operações internas a eles (PEREIRA, 2009, p. 27, tradução nossa).
45-46	27-2	Os dados de entrada são lidos a partir do <i>Coded Video Buffer</i> e enviados para o bloco <i>Serializer (Barrel Shifter)</i> , que alimenta os módulos <i>Exp-Golomb</i> , <i>Slice Data</i> e <i>Slice Header</i> em paralelo. De acordo com o estado na máquina de estados principal, os dados são interpretados e o número de bits de entrada utilizado no ciclo de relógio é realimentado para o <i>Serializer</i> para que no próximo ciclo de relógio o bit mais significativo no <i>Serializer</i>	Conforme explica Pereira (2009, p. 27, tradução nossa), os dados de entrada são lidos a partir do <i>Coded Video Buffer</i> e enviados para o bloco <i>Serializer (Barrel Shifter)</i> , que alimenta os módulos <i>Exp-Golomb</i> , <i>Slice Data</i> e <i>Slice Header</i> em paralelo. De acordo com o estado na máquina de estados principal, os dados são interpretados e o número de bits de entrada utilizado no ciclo de relógio é realimentado para o <i>Serializer</i> para que no próximo ciclo de relógio o

		seja o primeiro ainda não processado. A entrada é interpretada como um tipo de dados <i>signed integer</i> ou <i>unsigned integer</i> com o número de bits dependendo do símbolo sendo decodificado (código Exp-Golomb ou CAVLD) (DIEISON, 2009).	bit mais significativo no <i>Serializer</i> seja o primeiro ainda não processado. A entrada é interpretada como um tipo de dados <i>signed integer</i> ou <i>unsigned integer</i> com o número de bits dependendo do símbolo sendo decodificado (código Exp-Golomb ou CAVLD).
46	4	Em (STAEHLER, 2006)	No trabalho de Staehler (2006)
46	5	em (PEREIRA, 2009)	no trabalho de Pereira (2009)
46	9-14	Esta arquitetura é composta basicamente por dois algoritmos: a predição de luminância 4x4 e a de 16x16. Quando a predição intra-quadro 16x16 é utilizada, o modo de predição é codificado com o tipo de bloco. Para predição de crominância, 2 bits indicam o modo de predição. Para o bloco da predição intra 4x4, o modo de predição é adivinhado baseado nos blocos vizinhos. Este procedimento é chamado de <i>default mode</i> e utiliza 1 bit indicando se deve ou não ser utilizado.	Esta arquitetura é composta basicamente por dois algoritmos: a predição de luminância 4x4 e a de 16x16. Quando a predição intra-quadro 16x16 é utilizada, o modo de predição é codificado com o tipo de bloco. Para predição de crominância, 2 bits indicam o modo de predição. Para o bloco da predição intra 4x4, o modo de predição é adivinhado baseado nos blocos vizinhos. Este procedimento é chamado de <i>default mode</i> e utiliza 1 bit indicando se deve ou não ser utilizado (PEREIRA, 2009, p. 40, tradução nossa).
46	15-22	A arquitetura foi projetada para receber todas as informações de predição em paralelo, dessa forma, 1 bit é utilizado para indicar o tamanho do bloco (16 ou 4), 2 bits para o modo de predição 16x16, se usado, 16 bits indicando se cada um dos blocos de luminância 4x4 usa o	Ainda segundo Pereira (2009, p. 40, tradução nossa), a arquitetura foi projetada para receber todas as informações de predição em paralelo, dessa forma, 1 bit é utilizado para indicar o tamanho do bloco (16 ou 4), 2 bits para o modo de predição

		<p>modo de adivinhação, 48 bits para indicar o modo utilizado para blocos 4x4 quando o <i>default mode</i> não é utilizado e, finalmente, 2 bits indicam o modo de predição de crominância utilizado, somando-os, tem-se $1 + 2 + 16 + 48 + 2 = 69$ bits (PEREIRA 2009). Toda esta informação vem da FIFO de predição que conecta o módulo <i>parser</i> ao módulo de predição intra-quadro no decodificador <i>intra-only</i>.</p>	<p>16x16, se usado, 16 bits indicando se cada um dos blocos de luminância 4x4 usa o modo de adivinhação, 48 bits para indicar o modo utilizado para blocos 4x4 quando o <i>default mode</i> não é utilizado e, finalmente, 2 bits indicam o modo de predição de crominância utilizado, somando-os, tem-se $1 + 2 + 16 + 48 + 2 = 69$ bits. Toda esta informação vem da FIFO de predição que conecta o módulo <i>parser</i> ao módulo de predição intra-quadro no decodificador <i>intra-only</i>.</p>
46	23-25	<p>Uma vez que todos os dados são recebidos, a predição intra-quadro verifica o tamanho de bloco para predição, e então, define o modo de predição a ser utilizado, seleciona os vizinhos apropriados e executa os cálculos para o modo selecionado.</p> <p>Quatro amostras de 8 bits são processadas a cada ciclo de relógio, assim, após um atraso inicial, o módulo gera 96 amostras (4x8) de luminância ou crominância, formando assim um macrobloco predito de 384x8 bits.</p>	<p>Por fim, elucida Pereira (2009, p. 40-41, tradução nossa):</p> <p>uma vez que todos os dados são recebidos, a predição intra-quadro verifica o tamanho de bloco para predição, e então, define o modo de predição a ser utilizado, seleciona os vizinhos apropriados e executa os cálculos para o modo selecionado.</p> <p>Quatro amostras de 8 bits são processadas a cada ciclo de relógio, assim, após um atraso inicial, o módulo gera 96 amostras (4x8) de luminância ou crominância, formando assim um macrobloco predito de 384x8 bits.</p>
46	29	Em (AGOSTINI, 2007),	No trabalho de doutorado de Agostini (2007),
46	31	interligados e portanto	interligados, portanto

46	34	Em (PEREIRA, 2009),	No trabalho de Pereira (2009),
46	37-43	Nesta arquitetura, as saídas da FIFO que interconectam o módulo <i>parser</i> ao $Q^{-1}T^{-1}$ são decodificadas. O QP, o tamanho de bloco para predição intra-quadro e os resíduos são separados. Uma amostra é processada por ciclo de relógio, assim, 4 amostras são armazenadas para serem enviadas à FIFO que alimenta o IDCT. Os resíduos apresentam um <i>offset</i> em relação à imagem predita, corrigindo assim as distorções de predição. Como a predição varia de 0 a 255 (8 bits), os resíduos podem ter largura de 9 bits para serem capazes de cobrir toda a faixa de valores entre -255 e 255.	Nessa arquitetura, conforme explica Pereira (2009, p. 42, tradução nossa), as saídas da FIFO que interconectam o módulo <i>parser</i> ao $Q^{-1}T^{-1}$ são decodificadas. O QP, o tamanho de bloco para predição intra-quadro e os resíduos são separados. Uma amostra é processada por ciclo de relógio, assim, 4 amostras são armazenadas para serem enviadas à FIFO que alimenta o IDCT. Os resíduos apresentam um <i>offset</i> em relação à imagem predita, corrigindo assim as distorções de predição. Como a predição varia de 0 a 255 (8 bits), os resíduos podem ter largura de 9 bits para serem capazes de cobrir toda a faixa de valores entre -255 e 255.
47	22	estas <i>std-cells</i> já projetadas e verificadas podem	essas <i>std-cells</i> , já projetadas e verificadas, podem
47	25	Este fluxo	Esse fluxo
47	26	pré-definida, a fim	pré-definida a fim
47	30	alguns destes comerciais e extensivamente conhecidos, devido	alguns desses comerciais e extensivamente conhecidos devido
47	34	ainda outras	ainda outros
48	1	Além destes	Além desses
48	3	em (WAKABAYASHI, 2000)	no trabalho de Wakabayashi e Okamoto (2000)

48	5	Outra é discutida em (RICCOBENE, 2009)	Outra abordagem é discutida no trabalho de Riccobene et al. (2009)
48	11	em (BRANDON, 2005),	no trabalho de Brandon, Cockburn e Elliott (2005),
48	12	que além	que, além
48	30-31	porém, mapeada	porém mapeada
49	1	especificação, e por fim,	especificação e, por fim,
49	11	verificadas afim de saber	verificadas a fim de se averiguar
49	15	em (STAEHLER, 2006), em que estas	no trabalho de Staehler (2006), em que essas
49	19	FPGA, em trabalhos	FPGA em trabalhos
49	27	separadamente e gradativamente, de forma que para a	separada e gradativamente, de forma que, para a
49	29	e só após isso era	e, só após isso, era
49	36	para que este	para que o RTL
49	41-42	em Sreekandath et. al (SREEKANDATH, 1995)	no trabalho de Sreekandath e Priyadarshan (1995)
50	2	não haverão problemas	não haverá problemas
50	6-7	evitando assim perder muito tempo para corrigi-los depois que o projeto já está totalmente codificado.	evitando, assim, perder muito tempo para corrigi-los depois de o projeto estar totalmente codificado.
50	15	VHDL, trocar	VHDL, deve-se trocar
50	19	apesar das empresas	apesar de as empresas
50	33	o Apêndice pode	o Apêndice deste trabalho pode
50	35	FPGA, por macros	FPGA por macros

50	45	pois caso isso	pois, caso isso
51	2	lógica, devido	lógica devido
51	13	modificações, e a partir	modificações e a partir
51	26	e conseqüentemente funcionar	e, conseqüentemente, funcionar
51	28	até obter	até que se obtenha
52	1	e tentando assim descobrir	tentando, assim, descobrir
52	7	verificar isto:	verificar isso:
52	15	Existem ainda algumas	Existem, ainda, algumas
52	15	projeto, através	projeto por meio
52	17	Uma destas	Uma dessas
52	18	VeriSC (SILVA, 2006).	VeriSC (SILVA et al., 2006).
52	30	dados ou	dados, ou
53	8-9	e assim descobrir	e, assim, descobrir-se
53	14	lógica, ou	lógica ou
53	17	arquiteturas, e assim evitar	arquiteturas e, assim, evitar
53	28	sobre as erramentas	sobre as ferramentas
53	28	Apêndice pode	Apêndice deste trabalho pode
53	31	formal em que	formal, em que
53	34	Haas, et. al (HAAS, 2002)	no trabalho de Haas, Gossens e Heinkel (2002)
53	35	em (RODRIGUES, 2008),	no trabalho de Rodrigues et al. (2008)
53	42	implementação de todas	implementação física de todas
53	43	a cada vez que era	cada vez em que era
54	7	se execute sempre	se executem sempre

54	8	logicamente equivalente, e depois, a funcional,	logicamente equivalentes e, depois, a funcional,
54	13	projeto, em RTL, em	projeto feito RTL em
54	15	Esta descrição	Essa descrição
54	22	entrada e é necessário	entrada, e é necessário
54	29	etc., respectivas ao projeto	etc. do projeto;
54	35	outras ferramentas;	outras ferramentas.
54	42	que a RTL Compiler, além do fato de que para alguns	em relação ao RTL Compiler, além do fato de que, para alguns
55	1-2	o Apêndice.	o Apêndice deste trabalho.
55	7	no código original havia	havia no código original
55	16	que caso fosse	do que se fosse
56	3	versão 8.1, da Cadence.	versão 8.1 da Cadence.
56	21-22	camada 1, para	camada 1 para
56	23	no leiaute de forma	no leiaute, de forma
57	12	ocasionadas pela devido à irregularidade	ocasionadas pela irregularidade
57	13	no Apêndice	no Apêndice deste trabalho
57	14	<i>intra-only</i> em que foi inserido metal, após	<i>intra-only</i> na qual foi inserido metal após
57	17-18	também pode ser realizar separadamente, a fim	Também pode ser realizar separadamente a fim
57	23	é executada um	é executado um
57	25	etapas, e	etapas e
57	27	com análise	com a análise

57	29	leiaute e conseqüentemente afetando	leiaute e, conseqüentemente, afetando
57	34	conectado, e se é	conectada e se é
57	43	formal e funcional	formais e funcionais
58	3	certificar de	certificar-se de
59	7	em (ROSA, 2009)	no trabalho de Rosa, Bampi e Susin (2009),
59	8	aqui, pois no início deste trabalho de mestrado	neste trabalho de mestrado, pois, no início dele,
59	11	filtro, devido	filtro devido
60	3	física, realizada pro meio	física, realizada por meio
60	4	<i>Edge Filter</i> , do filtro	<i>Edge Filter</i> do filtro
60	6	sintetizados e juntos	sintetizados de forma conjunta
60	8	até chegar à disposição que proporcionou o melhor menor a	até chegar-se à disposição que proporcionou a menor
61	3	Encounter (CADENCE, 2010)	Encounter (CADENCE, 2010b)
61	3-4	CACTI (WILTON, 1996)	CACTI (WILTON; JOUPPI, 1996)
61	22-23	em (ROSA, 2009), implementada em ASIC	no trabalho de Rosa, Bampi e Susin (2009), na qual foi executado o fluxo ASIC, apresentado em Rosa, Silva e Bampi (2010) e
61	25-26	de (KIM, 2007)	da arquitetura apresentada no trabalho de Kim, Na e Kyung (2007)
61	27	Em (LIU, 2005)	No trabalho de Liu et al. (2005)
61	35	de (KIM, 2007) e com a implementação ASIC de (ROSA, 2009)	de Kim, Na e Kyung (2007) e com os resultados do fluxo de implementação ASIC apresentados em Rosa, Silva e Bampi (2010) para a arquitetura de Rosa, Bampi e Susin (2009)

62	Tab. 5.4	(LIU, 2005) (SHIH, 2006) (KIM, 2007) (ROSA, 2009)	Liu et al. (2005) Shi, Chang e Lin (2006) Kim, Na e Kyung (2007) Rosa, Bampi e Susin (2009)
62	9	de (SHIH, 2006)	Shi, Chang e Lin (2006)
62	17	de (KIM, 2007)	Kim, Na e Kyung (2007)
62	23	de (ROSA, 2009),	da arquitetura apresentada no trabalho de Rosa, Bampi e Susin (2009)
62	27	em (ROSA, 2009)	na arquitetura apresentada no trabalho de Rosa, Bampi e Susin (2009)
63	1	que para	que, para
63	7-8	trabalho, implementar em ASIC, não só uma	trabalho executar o fluxo de implementação ASIC não só em uma
63	9-10	em (AZEVEDO, 2006)	no trabalho de Azevedo et al. (2007)
63	11	em (ZATT, 2008a)	no trabalho de Zatt (2008a)
63	12	foi implementada em ASIC	passou pelo fluxo de implementação ASIC
63	15-16	foi também implementada em ASIC	também passou pelo fluxo de implementação ASIC
63	19-20	<i>Main</i> (ZATT, 2010a), também foi inicialmente implementado separadamente, com o intuito	<i>Main</i> (ZATT et al., 2010a), inicialmente também foi implementado separadamente com o intuito
63	21	Após o MVP ter sido implementado	Após ter sido executado o fluxo de implementação física no MVP
63	25	MVP implementado em ASIC e corresponde	MVP, que passou pelo fluxo de implementação ASIC, corresponde
64	1-2	em (ZATT, 2010a), e implementada em ASIC neste trabalho,	no trabalho de (ZATT et al., 2010a), a qual passou pelo fluxo de implementação ASIC neste trabalho,
64	4-13	Entre as arquiteturas de hardware para MVP	Nesse contexto, é explicado no trabalho de Zatt (2010a, p. 18, tradução nossa):

		<p>encontradas, tanto (WANG, 2005) como (XU, 2008) apresentam soluções para o perfil <i>Baseline</i> do H.264/AVC. A principal diferença entre elas e o MVP perfil <i>Main</i> implementado em ASIC aqui, é que o perfil <i>Baseline</i> suporta apenas quadros do tipo P, e conseqüentemente estas arquiteturas não possuem gerenciamento da lista 1 de predição, ou predição direta. Devido a estas restrições e à baixa resolução de vídeo (QCIF - 176x144 pixels) suportada por (XU, 2008), ela apresenta um menor número de <i>gates</i> em relação à implementação em ASIC deste trabalho de mestrado, que consegue decodificar vídeos com resolução HD 1080p@30fps executando em frequências menores que as outras (ZATT, 2010a).</p>	<p>Entre as arquiteturas de hardware para MVP encontradas, tanto o trabalho de Wang, S. et al. (2005) como o de Xu e Choy (2008) apresentam soluções para o perfil <i>Baseline</i> do H.264/AVC. A principal diferença entre elas e o MVP proposto por Zatt é que o perfil <i>Baseline</i> suporta apenas quadros do tipo P e, conseqüentemente, essas arquiteturas não possuem gerenciamento da lista 1 de predição, ou predição direta. Devido a essas restrições e à baixa resolução de vídeo (QCIF - 176x144 pixels) suportada no trabalho de Xu e Choy (2008), ela apresenta um menor número de <i>gates</i> em relação à arquitetura proposta por Zatt, que consegue decodificar vídeos com resolução HD 1080p@30fps executando em frequências menores que as outras.</p>
64	Tab. 5.6	(WANG, 2005) (XU, 2008) (CHEN, 2006) (ZHENG, 2008)	(WANG, S. et al., 2005) (XU; CHOY, 2008) (CHEN, J. et al., 2006) (ZHENG et al., 2008)
64	17	trabalho de (CHEN, 2006)	trabalho de Chen, J. et al. (2006)
64	20-27	Por fim, em (ZHENG, 2008), é apresentada a única solução em hardware encontrada, que suporta todas as características do perfil <i>Main</i> do H.264/AVC, incluindo predição direta espacial e temporal. Ela também foi projetada para	Com relação aos resultados do fluxo de implementação física executado neste trabalho para a arquitetura proposta por Zatt (2010a), ressalta-se a comparação de Zatt (2010a, p18. Tradução nossa) em relação à arquitetura proposta por Zheng et al. (2008): no trabalho de Zheng et al.

		<p>predizer vetores de movimento de acordo com os padrões MPEG-2 e AVS, e possui 8K <i>gates</i> a menos que a implementação ASIC realizada neste trabalho, o que pode ser justificado pelo <i>throughput</i> de (ZHENG, 2008), uma vez que para a arquitetura de (ZHENG, 2008), é necessário uma frequência de operação maior, de 148 MHz, para decodificar vídeos HD 1080p@30fps.</p>	<p>(2008), é apresentada a única solução em hardware encontrada que suporta todas as características do perfil <i>Main</i> do H.264/AVC, incluindo predição direta espacial e temporal. Ela também foi projetada para predizer vetores de movimento de acordo com os padrões MPEG-2 e AVS.</p> <p>A arquitetura proposta em Zheng et al. (2008) possui 8K <i>gates</i> em comparação com os resultados do fluxo de implementação física encontrados neste trabalho de mestrado em relação à arquitetura proposta por Zatt (2010a) e, segundo justificativa de Zatt (2010a, p. 18, tradução nossa), essa diferença pode ser justificada</p> <p>pelo <i>throughput</i> apresentado na arquitetura proposta por Zheng et al. (2008), uma vez que, para essa arquitetura, é necessário uma frequência de operação maior, de 148 MHz, para decodificar vídeos HD 1080p@30fps.</p>
64	31	MC perfil <i>Main</i>	MC MoCHA (AZEVEDO, 2007) perfil <i>Main</i>
65	5	MC HP422-MoCHA	MC HP422-MoCHA (Zatt, 2008a)
66	3	MC HP422-MoCHA	MC HP422-MoCHA (Zatt, 2008a)
66	4	MC MoCHA	MC MoCHA (AZEVEDO, 2007)
67	1-2	MC MoCHA, feita anteriormente, reside	MC MoCHA (AZEVEDO, 2007) perfil <i>main</i> feita anteriormente reside
67	3	<i>High</i> 4:2:2, e também	<i>High</i> 4:2:2 e também
67	8	Apesar do MVP tem	Apesar do MVP ter

67	10	área, devido à grande quantidade de bancos de memória isntanciados	área devido à grande quantidade de bancos de memória instanciados
67	15	na implementação do filtro	na execução do fluxo de implementação física do filtro
67	27	da impemlentação	da implementação
67	28-29	em (FINCHELSTEIN, 2008)	no trabalho de et al. Finchelstein (2008)
67	31	Analisando a figura,	Analisando-se a Figura 5.6,
67	34	potência .em	potência em
68	5-6	de (WANG, 2005), (WANG R., 2005) e (XU, 2008)	de Wang, S. et al, (2005), Wang, R. (2005) e Xu e Choy (2008)
68	6-7	de (CHEN, 2006) e (ZHENG, 2008)	de Chen, J. et al. (2006) e Zheng et al. (2008)
68	11-12	do MC MoCHa e HP422-MoCHA, e outras	do MC MoCHa (AZEVEDO, 2007) e HP422-MoCHA (Zatt, 2008a) e outras
68	17-18	interpolador, enquanto a segunda, a	interpolador enquanto a segunda corresponde a
68	19	de (WANG, 2005)	de Wang, S. et al, (2005)
68	27	de (WANG, 2005)	de Wang, S. et al, (2005)
68	29	de (WANG R., 2005), é	de Wang, R. (2005) é
68	30	ser necessário 492	serem necessário 492
69	Tab. 5.10	(WANG, 2005) (WANG R., 2005) (XU, 2008) (CHEN, 2006) (ZHENG, 2008)	(WANG, S. et al, 2005) (WANG, R. et al., 2005) (XU; CHOY, 2008) (CHEN et al, 2006) (ZHENG et al., 2008)
69	Tab. 5.10	<i>Hierarchy de Mem. Gate Count</i>	<i>Mem Hierarchy Gate Count</i>
69	8	em (XU, 2008)	em Xu e Choy (2008)
69	17	apresentado (CHEN, 2006)	de Chen, J. et al. (2006)

69	20	de (CHEN, 2006)	de Chen, J. et al. (2006)
70	3	Em (ZHENG, 2008)	No trabalho de Zheng et al. (2008)
70	11	a 30fps	a 30fps (ZATT et al., 2010a)
70	18	trabalhos (ZATT, 2008b)	trabalhos (ZATT et al., 2010a)
70	20-21	<i>intra-only</i> apresentado em é	<i>intra-only</i> é
70	23	Todas essas foram	Todas essas arquiteturas foram
70	24	de (PEREIRA, 2006)	de Pereira (2009)
70	25	Neste trabalho, foi realizada a implementação física (ASIC) deste decodificador, utilizando	Neste trabalho, foi executado o fluxo de implementação física (ASIC) para esse decodificador, conforme apresentado por este autor em Bonatto et al. (2010) e Zatt et al. (2010b), utilizando
70	27	arquiteturas do filtro	arquiteturas filtro
70	28	a implementação física	a execução do fluxo de implementação física
70	32	entropia, e por fim,	entropia e, por fim,
73	3	literatura apresentado	literatura apresentando
73	4	H.264 <i>intra-only</i>	H.264 <i>intra-only</i> gerado no trabalho de Pereira (2009)
73	5-6	da implementação em ASIC desde decodificador <i>intra-only</i> , com	dos resultados do fluxo de implementação física (ASIC) do decodificador decodificador <i>intra-only</i> , apresentados por este autor em Bonatto et al. (2010) e Zatt et al. (2010b), com
73	Tab. 5.13	(NA, 2007) (CHEN T., 2006) (LIN, 2007)	(NA et al., 2007) (CHEN, T; LIAN; CHEN, L., 2006) (LIN et al., 2007)
73	11	Em (CHEN T., 2006)	No trabalho de Chen, T., Lian e Chen, L (2006)
73	13	resolução, ao custo	resolução ao custo

73	15	de (LIN, 2007)	de Lin et al. (2007)
74	2	potência, com	potência com
74	4	de (NA, 2007)	de Na et al. (2007)
74	6	(<i>intra-only</i>), e	(<i>intra-only</i>) e
74	7-8	ao decodificador implementado H.264 em ASIC	aos resultados gerados para o decodificador H.264 <i>intra-only</i> usado
75	2	a implementação física (ASIC, em <i>standard-cells</i>) das arquiteturas	os resultados do fluxo de implementação física (ASIC, em <i>standard-cells</i>), expostos por este autor em Rosa, Silva e Bampi (2010), Zatt et al. (2010a), Silva, Pereira e Bampi (2010), Bonatto et al. (2010) e Zatt et al. (2010b), para as seguintes arquiteturas
75	6	H.264 <i>intra-only</i>	H.264 <i>intra-only</i> gerado no trabalho de Pereira (2009)
75	8-9	Todas as arquiteturas foram fisicamente implementadas utilizando	Para todas as arquiteturas foi executado o fluxo de implementação física utilizando
75	10	resumida, o padrão	resumida o padrão
75	11-12	a serem implementadas em ASIC, e em seguida,	em que seria executado o fluxo de implementação ASIC e, em seguida,
75	18	que para	que, para
75	19	chegar até	chegar-se até
75	22	código, rodadas	código e rodadas
75	27-28	codificação deste seja	codificação dele seja
75	30	ASIC, após	ASIC após
75	34-35	visa implementação tanto em FPGA quanto ASIC, é aconselhável que seguir	visa a implementação tanto em FPGA quanto em ASIC, é aconselhável seguir
76	1	considerar, é a	considerar é a

76	3-4	desenvolvedor/projetista, responsável pela codificação de projeto de hardware para ASIC, tenha	projetista responsável pela codificação de projeto de hardware para ASIC tenha
76	5	permitindo-o assim, ficar	permitindo-o, assim, ficar
76	9	Todas as arquiteturas de hardware mencionadas foram implementadas em ASIC, estão disponíveis como resultados gerais do trabalho e podem,	As arquiteturas de hardware mencionadas acima passaram pelo fluxo de implementação física (ASIC) e seus resultados estão disponíveis como resultados gerais deste trabalho de mestrado podendo,
76	11	H.264/AVC, a ser	H.264/AVC a ser
76	17	como VeriSC	como a VeriSC (SILVA et al., 2006)
76	19	arquiteturas, alguma	arquiteturas alguma
76	23	tornando-o assim,	tornando-o, assim,
76	26-27	filtro, visando	filtro visando
76	29	provavelmente estão	provavelmente estarão
76	32	arquiteturas, utilizando	arquiteturas utilizando
76	33	visando assim, obter	visando, assim, obter
76	35	literatura que utilizam	literatura e que utilizam
76	37	ele, e este	ele e este
85	15-16	Quartus 2 (ALTERA, 2010)	Quartus 2 (ALTERA, 2010a)
93	25	Specman/e (CADENCE, 2010).	Specman/e (CADENCE, 2010a).
93	31	VeriSC (SILVA, 2006).	VeriSC (SILVA et al., 2006).
94	15	VeriSC (SILVEIRA, 2009)	VeriSC (SILVEIRA; BRITO; MELCHER, 2009, p. 4).
94	28	MPEG-4 (ROCHA, 2006),	MPEG-4 (ROCHA et al., 2006),

103	9	em (ALTERA, 2010)	em (ALTERA, 2010b)
103	43	CDD (CADENCE, 2010)	CDD (CADENCE, 2010a)
133	46-47	PowerMeter (CADENCE, 2010)	PowerMeter (CADENCE, 2010a)
134	7	VoltageStorm (CADENCE, 2010)	VoltageStorm (CADENCE, 2010a)
Folha	Linha	REFERÊNCIAS BIBLIOGRÁFICAS	
77	19	Mover a referência “BHASKARAN, V.; KONSTANTINIDES, K. [...]” para a posição em ordem alfabética.	
77	19	Adicionar a referência “BONATTO, A. C. ; SOARES, A. B. ; SUSIN, A. A. . Controlador DDR SDRAM Multicanal de Alta Velocidade Aplicado à Decodificação H.264/AVC. In: Workshop Iberchip, 2010, Foz do Iguaçu, Brasil. Workshop Iberchip, 2010.”	
78	7	Onde se lê “CHEN, J. W et al.” leia-se “CHEN, J. et al.”	
79	19	Onde se lê “Filter Architecture”, leia-se “Filter Architecture”	
79	22	Mude a referência “LIU, T. M. et al. [...]” para a posição em ordem alfabética.	
80	13	Onde se lê “PORTO, B. E. C”, leia-se “PORTO, R. E. C.”	
80	31	Onde se lê “ROSA, V. S. et al.” leia-se “ROSA, V. S.; SILVA, L. M. de L.; Bampi, S.”	
81	1	Mover a referência “SHI, Y.; SUN, H. [...]” para a posição em ordem alfabética.	
81	11	Onde se lê “SILVEIRA, G. S. et al.”, leia-se “SILVEIRA, G. S.; BRITO, A. V; MELCHER, E. U. K.”	
81	24	Adicionar a referência “SULLIVAN, G. et al. The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions. In: CONFERENCE ON APPLICATIONS OF DIGITAL IMAGE PROCESSING, 27., 2004. Proceedings...”	

		[S.l.:s.n.], 2004.”
82	3	Onde se lê “WANG, R; LI, M.; LI, J.; ZHANG, Y”, leia-se “WANG, R. et al”