

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

CLÁUDIO MACHADO DINIZ

**Arquitetura de Hardware Dedicada para a
Predição Intra-Quadro em Codificadores do
Padrão H.264/AVC de Compressão de Vídeo**

Dissertação apresentada como requisito parcial
para a obtenção do grau de Mestre em Ciência
da Computação

Prof. Dr. Altamiro Amadeu Susin
Orientador

Porto Alegre, setembro de 2009.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Diniz, Cláudio Machado

Arquitetura de Hardware Dedicada para a Predição Intra-Quadro em Codificadores do Padrão H.264/AVC de Compressão de Vídeo / Cláudio Machado Diniz – Porto Alegre: Programa de Pós-Graduação em Computação, 2009.

96 p.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2009. Orientador: Altamiro Amadeu Susin.

1. Compressão de vídeo. 2. H.264/AVC 3. Predição Intra-Quadro 4. Hardware. I. Susin, Altamiro Amadeu. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Pós-Graduação: Prof. Aldo Bolten Lucion

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do PPGC: Prof. Álvaro Freitas Moreira

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Uma dissertação não se faz sozinho. Gostaria de agradecer a todas as pessoas que estiveram envolvidas e me ajudaram no decorrer do desenvolvimento deste trabalho, esperando que elas possam compartilhar junto comigo a conquista de mais esta etapa.

Agradeço de coração à minha namorada, Cilene, por sua compreensão com a minha ausência, minha falta de atenção e meu mau humor em diversos momentos. Obrigado por ter me ajudado desde o início, na mudança para Porto Alegre, mesmo sabendo que ficaríamos longe a maior parte do tempo durante estes longos dois anos. Obrigado por me incentivar e não me deixar desistir nunca, diante de todas as dificuldades que encontrei. Agradeço pelo teu carinho comigo, pelas conversas, e por estares ao meu lado em todos os momentos. Espero poder recompensá-la por tudo isto. Eu te amo!

À minha mãe, Eliane, que me transmitiu toda a educação e valores que carrego comigo até hoje. Obrigado por ter me ajudado sempre que eu precisei, nunca deixando que eu passasse necessidade. Agradeço pelas palavras de apoio, que sempre me estimularam para eu acreditar no que faço e seguir em frente. Obrigado também por ser, além de uma mãe maravilhosa, uma profissional na qual eu tento me espelhar: incansável, responsável e honesta acima de tudo.

Ao meu irmão, Fernando, pela companhia durante estes dois anos que moramos juntos em Porto Alegre. Foram poucos os momentos juntos, devido aos nossos horários desencontrados, mas que sempre renderam conversas muito proveitosas.

Ao meu orientador, Prof. Altamiro Susin, por ter me aceitado no mestrado, pelas orientações e ensinamentos, por ter acreditado no meu trabalho e por ter compreendido as minhas falhas em respeitar prazos. Agradeço também ao Prof. Sergio Bampi, pelos ensinamentos e por estar sempre disposto a me orientar e revisar meus trabalhos, mesmo não sendo meu orientador formalmente.

Ao Prof. Luciano Agostini, pelas orientações e revisões de artigos e, mais importante, por ter me ajudado a mudar o assunto da minha dissertação, fato este que foi primordial para definir o rumo deste e de outros trabalhos que virão em breve.

Ao pessoal do meu grupo de pesquisa, o “lab215”, por se mostrarem ótimos companheiros durante estes anos, seja nos trabalhos em conjunto, nas conversas filosóficas da hora do cafezinho e nos churrascos e festas da turma. Obrigado especialmente ao Vagner, que me convenceu a vir trabalhar no grupo de pesquisa dele, ao Bruno, com quem divido autoria em alguns trabalhos, ao Dieison, Marcelo, Roger, e Thaísa, que me ajudaram muito na adaptação inicial do mestrado, ao Guilherme Mauch, que foi meu colega de vários trabalhos, ao Fábio, Débora, Leandro Zanetti, Leandro Max, Guilherme Corrêa, André e Miklécio, que entraram depois e trouxeram mais alegria e competência ao nosso grupo de pesquisa.

Ao pessoal do GACI da UFPel, em especial ao Felipe, Robson “Duda” e Daniel, por serem tão ágeis e dedicados para me auxiliar nas tarefas de integração e verificação com os módulos de transformadas/quantização desenvolvidos por eles. Devo um agradecimento especial ao Duda, por ter vindo de Pelotas nas férias para me ajudar na tarefa de verificação.

Aos professores do PPGC que contribuíram para minha formação: Sergio Bampi, Altamiro Susin, Flávio Wagner, Luigi Carro, Fernanda Kastensmidt e Marcelo Lubaszewski.

Aos meus amigos, colegas e parentes, não citados nominalmente, mas que proporcionaram boas conversas, risadas e momentos de descontração, obrigado.

Ao CNPq, por financiar meus dois anos de mestrado através de uma bolsa de estudos.

A todos, meu muito obrigado!

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	9
LISTA DE FIGURAS	13
LISTA DE TABELAS	15
RESUMO	17
ABSTRACT	19
1 INTRODUÇÃO	21
2 CONCEITOS DE COMPRESSÃO DE VÍDEO E O PADRÃO H.264/AVC .	25
2.1 Captura e Representação do Vídeo Digital	25
2.1.1 Captura.....	25
2.1.2 Espaços de cores e sub-amostragem de cores.....	25
2.1.3 Métricas para Avaliação de Qualidade	26
2.2 Conceitos Básicos de Compressão de Vídeo	27
2.2.1 Técnicas de Compressão com Perdas de Informação	27
2.2.2 Redundâncias de Dados na Representação de Vídeos.....	27
2.3 Introdução ao padrão H.264/AVC	28
2.3.1 Histórico do Padrão H.264/AVC	28
2.3.2 Estrutura de Codificação.....	29
2.3.3 Perfis e Níveis	30
2.3.4 Formato do Vídeo Codificado	32
2.4 Núcleo do Codec H.264/AVC	32
2.4.1 Estimação de Movimento (ME).....	35
2.4.2 Compensação de Movimento (MC).....	37
2.4.3 Transformadas Diretas (T).....	38
2.4.4 Quantização Direta (Q).....	39
2.4.5 Quantização Inversa (QI).....	40
2.4.6 Transformadas Inversas (TI).....	41
2.4.7 Filtro Redutor de Efeitos de Bloco	42
2.4.8 Codificação de Entropia.....	43
3 A PREDIÇÃO INTRA-QUADRO	45
3.1 Técnicas de Exploração da Redundância Espacial	45
3.2 A Predição Intra-Quadro no H.264/AVC	46

3.2.1	Modos de Predição Intra 4x4	48
3.2.2	Modos de Predição Intra 16x16	50
3.2.3	Modos de Predição para Crominância	51
3.2.4	Cálculo da Disponibilidade dos Blocos Vizinhos.....	52
3.2.5	Sinalização dos Modos de Predição Intra-Quadro no Vídeo Codificado	54
3.3	Processo de Codificação Intra-Quadro.....	56
4	ARQUITETURA PARA PREDIÇÃO INTRA-QUADRO DO PADRÃO	
	H.264/AVC	59
4.1	Investigação Arquitetural para Predição Intra-Quadro.....	59
4.2	Arquitetura para Predição Intra-Quadro	61
4.2.1	Gerenciamento das Amostras Vizinhas e Máquina de Controle	62
4.2.2	Preditor de Amostras	64
4.2.3	Cálculo do SAD	69
4.2.4	Decisão do Modo de Predição	70
4.2.5	<i>Buffer</i> e Memória de Predição	71
4.2.6	Cálculo de Resíduos e Reconstrução	73
4.3	Integração com os Módulos de Transformadas e Quantizações Diretas e	
	Inversas.....	74
4.4	Resultados de Síntese.....	74
4.5	Análise de Desempenho	77
4.6	Metodologia de Verificação.....	80
5	COMPARAÇÕES COM TRABALHOS RELACIONADOS.....	83
5.1	Revisão de Trabalhos da Literatura	83
5.1.1	Trabalho de Huang et al.	83
5.1.2	Trabalho de Suh, Park e Cho	84
5.1.3	Trabalho de Cheng, Ku e Chang.....	84
5.1.4	Trabalho de Ku et al.....	85
5.1.5	Trabalho de Li et al.	85
5.1.6	Trabalho de Chang et al.	85
5.1.7	Trabalho de Sahin e Hamzaoglu	85
5.1.8	Trabalho de Kuo e Lin	86
5.1.9	Trabalho de Lin et al.	86
5.2	Comparações com a Arquitetura Desenvolvida.....	86
6	CONCLUSÕES E TRABALHOS FUTUROS.....	89
6.1	Trabalhos Futuros	90
	REFERÊNCIAS.....	93

LISTA DE ABREVIATURAS E SIGLAS

1-D	Unidimensional
2-D	Bidimensional
ASIC	<i>Application Specific Integrated Circuit</i>
ASM	<i>Algorithmic State Machine</i>
AVC	<i>Advanced Video Coding</i>
BAC	<i>Binary Arithmetic Coding</i>
BITSTREAM	Sequência de bits do vídeo codificado
BRAM	Block RAM
CABAC	<i>Context-Based Adaptive Binary Arithmetic Coding</i>
CAVLC	<i>Context-Based Adaptive Variable Length Coding</i>
Cb	<i>Chrominance Blue</i>
CCD	<i>Charge Coupled Device</i>
CBP	<i>Coded Block Pattern</i>
CIF	<i>Common Intermediate Format</i>
codec	Codificador/Decodificador
Cr	<i>Chrominance Red</i>
DCT	<i>Discrete Cosine Transform</i>
D1	<i>Standard Definition</i>
DVD	<i>Digital Versatile Disk</i>
FRExt	<i>Fidelity Range Extensions</i>
FSM	<i>Finite State Machine</i>
GACI	Grupo de Arquiteturas e Circuitos Integrados
GB	<i>Gigabytes</i>
HD	<i>High Definition</i>
HDTV	<i>High Definition Digital Television</i>
IDR	<i>Instantaneous Decoding Refresh</i>
IEC	<i>International Electrotechnical Commission</i>

INTRA	Predição Intra-Quadro
INTER	Predição Inter-Quadros
ISO	<i>International Organization for Standardization</i>
ITU	<i>International Telecommunication Union</i>
I4MB	Modo de predição Intra 4x4
I16MB	Modo de predição Intra 16x16
JPEG	<i>Joint Photographic Experts Group</i>
JVT	<i>Joint Video Team</i>
LUT	<i>Look-up Table</i>
ME	<i>Motion Estimation</i>
MC	<i>Motion Compensation</i>
MPEG	<i>Motion Picture Experts Group</i>
MSE	<i>Mean-Squared Error</i>
NAL	<i>Network Abstraction Layer</i>
POC	<i>Picture Order Count</i>
PSNR	<i>Peak Signal-to-Noise Ratio</i>
RAM	<i>Random Access Memory</i>
RDO	<i>Rate-Distortion Optimization</i>
RGB	<i>Red, Green, Blue</i>
QCIF	<i>Quarter Common Intermediate Format</i>
QI	Quantização Inversa
QP	<i>Quantization Parameter</i>
Qstep	<i>Quantization Step</i>
SAD	<i>Sum of Absolute Differences</i>
SATD	<i>Sum of Absolute Transformed Differences</i>
SDTV	<i>Standard Television</i>
SVC	<i>Scalable Video Coding</i>
SI	<i>Switch-I</i>
SP	<i>Switch-P</i>
VCEG	<i>Video Coding Experts Group</i>
VCL	<i>Video Coding Layer</i>
VGA	<i>Video Graphics Array</i>
VLC	<i>Variable Length Coding</i>
VHDL	<i>VHSIC Hardware Description Language</i>
VHSIC	<i>Very High Speed Integrated Circuits</i>

TI	Tranformadas Inversas
UFPel	Universidade Federal de Pelotas
UFRGS	Universidade Federal do Rio Grande do Sul
Y	<i>Luminance</i>
YCbCr	<i>Luminance, Chrominance Blue, Chrominance Red</i>

LISTA DE FIGURAS

Figura 2.1: Perfis do H.264/AVC.....	31
Figura 2.2: Diagrama em blocos de um codificador H.264/AVC.....	33
Figura 2.4: Diagrama em blocos de um decodificador H.264/AVC.....	34
Figura 2.4: Determinação do vetor de movimento para um bloco.....	36
Figura 2.5: Divisão do macrobloco em partições de macrobloco.....	37
Figura 2.6: Divisão do sub-macrobloco em partições.....	37
Figura 2.7: Ordem de processamento das amostras no bloco T.....	38
Figura 2.8: Ordem de filtragem das bordas de um macrobloco.....	42
Figura 2.9: Amostras adjacentes para bordas verticais e horizontais.....	43
Figura 3.1: Tipos de macrobloco Intra-Quadro.....	46
Figura 3.2: Detalhe da aplicação da predição Intra 4x4 em um quadro da sequência de vídeo <i>Foreman</i> (QCIF).....	47
Figura 3.3: Resultado da predição Intra 4x4. Quadro original (esquerda); Resíduo da predição Intra 4x4 (direita).....	47
Figura 3.4: Identificação das amostras para a predição Intra 4x4.....	48
Figura 3.5: Modos de predição Intra 4x4.....	48
Figura 3.6: Modos de predição Intra 16x16.....	50
Figura 3.7: Modos de predição para blocos 8x8 de crominância.....	51
Figura 3.8: Convenção para indicar macroblocos ou blocos vizinhos.....	52
Figura 3.9: Processo de codificação intra-quadro.....	56
Figura 4.1: Primeira solução arquitetural para a predição intra-quadro.....	60
Figura 4.2: Segunda solução arquitetural para a predição intra-quadro.....	60
Figura 4.3: Diagrama de tempo comparando as duas soluções arquiteturais.....	60
Figura 4.4: Arquitetura proposta e desenvolvida a predição intra-quadro e sua relação com os demais módulos que compõem um codificador de vídeo somente para quadros Intra.....	62
Figura 4.5: Amostras reconstruídas (em cinza) armazenadas em memória interna para predição.....	63
Figura 4.6: Diagrama das máquinas de controle. a) FSM de controle e leitura das amostras vizinhas; b) FSM de armazenamento das amostras vizinhas.....	64
Figura 4.7: Arquitetura do preditor para modos diagonais (I4MB) usando decomposição de equações.....	65
Figura 4.8: Arquitetura do preditor para modos diagonais (I4MB) usando eliminação de sub-expressões comuns.....	66
Figura 4.9: Modo Plano I16MB Usando Multiplicadores.....	68
Figura 4.10: Modo Plano I16MB Usando Apenas Somas e Deslocamentos.....	68
Figura 4.11: Arquitetura para cálculo do SAD.....	69
Figura 4.12: Arquitetura para Decisão do Modo de Predição I4MB.....	70

Figura 4.13: Arquitetura para Decisão do Modo de Predição I16MB.....	71
Figura 4.14: Arquitetura para Decisão do Modo de Predição Croma.	71
Figura 4.15: Organização das Memórias de Predição para Intra 16x16 e Crominância.	73
Figura 4.16: Interface Entre o Preditor Intra-Quadro e o Loop TQ/IQIT.	74
Figura 4.17: Processamento intercalado Intra 4x4 (I4MB) / Intra 16x16 (I16MB).	78
Figura 4.18: Restante do processo de codificação intra-quadro.	79
Figura 4.19: Resultados do <i>software</i> de predição intra-quadro para um quadro da sequência de vídeo 'Foreman' (QCIF).	81
Figura 4.20: Resultados de saída das arquiteturas de predição intra-quadro, transformadas e quantização diretas e inversas (somente I4MB) para um quadro da sequência de vídeo 'Foreman' (QCIF) em diferentes parâmetros de quantização (QP).	82

LISTA DE TABELAS

Tabela 2.1: Níveis do H.264/AVC	32
Tabela 2.2: Relação entre QP e Qstep	39
Tabela 2.3: Dez primeiros códigos <i>Exp-Golomb</i>	44
Tabela 3.1: Modos de Predição Intra 4x4 (I4MB).....	49
Tabela 3.2: Tipos de macrobloco para <i>slices</i> I	54
Tabela 3.3: Escolha do modo Intra 4x4 (modo mais provável = 1).	56
Tabela 4.1: Operações Necessárias para Cálculo dos Modos de Diagonais (I4MB).	65
Tabela 4.2: Comparação Entre Soluções para Cálculo dos Modos Diagonais (I4MB). 67	
Tabela 4.3: Resultados de Síntese para FPGA Xilinx Virtex-II Pro.	75
Tabela 4.4: Resultados de Síntese para FPGA Xilinx Virtex-5.	76
Tabela 4.5: Resultados de Síntese para <i>Standard-Cells</i> TSMC 0.18 μ m.	77
Tabela 4.6: Desempenho da arquitetura para diversas resoluções.	80
Tabela 5.1: Resultados comparativos em termos de uso de recursos de <i>hardware</i>	87
Tabela 5.2: Resultados comparativos em termos de desempenho.....	87

RESUMO

A compressão de vídeo é essencial para aplicações de vídeo digital. Devido ao elevado volume de informações contidas em um vídeo digital, um processo de compressão é aplicado antes de ser armazenado ou transmitido. O padrão H.264/AVC é considerado o estado-da-arte em termos de compressão de vídeo, introduzindo um conjunto de ferramentas inovadoras em relação a padrões anteriores. Tais ferramentas possibilitam um ganho significativo em compressão, ao preço de um aumento na complexidade.

A predição intra-quadro é uma das ferramentas inovadoras do padrão H.264/AVC, responsável por reduzir a redundância espacial do vídeo utilizando informações contidas em um único quadro para predição. A predição intra-quadro do H.264/AVC possibilita ganhos de compressão em comparação com os mais usados padrões de compressão de imagens estáticas, o JPEG e JPEG 2000, mas introduz complexidade no projeto do codificador de vídeo, especialmente quando se torna necessário atingir o desempenho para codificar vídeos de alta definição em tempo-real.

Neste contexto, a presente dissertação apresenta a proposta e o desenvolvimento de uma arquitetura de *hardware* dedicada para a predição intra-quadro, presente nos codificadores compatíveis com o padrão H.264/AVC de compressão de vídeo. A arquitetura desenvolvida codifica vídeos de alta definição em tempo-real utilizando uma frequência de operação 46% menor que o melhor trabalho encontrado na literatura. A arquitetura desenvolvida será integrada, futuramente, em um codificador de vídeo em *hardware* compatível com o padrão H.264/AVC no perfil *Main*.

Palavras-Chave: Compressão de Vídeo, H.264/AVC, Predição Intra-Quadro, Projeto de *Hardware* Digital, Vídeo de Alta Definição.

Intra-Frame Prediction Dedicated Hardware Architecture for Encoders of the H.264/AVC Video Coding Standard

ABSTRACT

Video coding is essential in digital video applications, due to the extremely high data volume present in a digital video to be stored or transmitted through a physical link. H.264/AVC is the state-of-the-art video coding standard, introducing a set of novel features when compared to former standards. A significant gain in terms of bit-rate has been obtained but the increase of complexity of the codec when compared to other video coding standard is inevitable.

Intra-frame Prediction is a novel feature introduced with H.264/AVC, which is responsible for reducing a video spatial redundancy using only information in the same frame for prediction. H.264/AVC intra-frame prediction can provide compression gains when compared with state-of-art still image coding standards, like JPEG and JPEG 2000, but introduces complexity and latency to video encoder design, mainly when high definition video coding is needed.

In this context, this thesis presents the proposal and development of an intra-frame prediction dedicated hardware architecture for H.264/AVC compatible video encoder. The developed architecture achieved the performance to encode high definition video in real-time with 46% reduction in clock frequency compared with the best results found in the literature. In the future, the developed architecture can be integrated to a fully compatible H.264/AVC main profile hardware encoder.

Keywords: Video Coding, H.264/AVC, Intra-Frame Prediction, Digital Hardware Design, High Definition Video.

1 INTRODUÇÃO

Atualmente, existe no mundo uma crescente proliferação de aplicações de vídeo digital, seja em dispositivos móveis, tais como celulares, câmeras digitais, *players* portáteis de vídeo, ou em dispositivos fixos, como os vídeos pela Internet e os televisores digitais. O início da transmissão de televisão digital no Brasil, iniciada oficialmente no dia 3 de dezembro de 2007, contribuiu significativamente para o aumento da demanda por dispositivos que processam vídeo digital.

A compressão de vídeo é necessária devido ao elevado volume de dados presentes nos vídeos digitais. Um vídeo com resolução de 720x480 pixels a 30 quadros por segundo (usado em televisão digital com definição normal – SDTV, assim como em DVDs) e utilizando 24 bits por pixel necessitaria de uma taxa de transmissão próxima a 249 Mbps. Um trecho com 10 minutos de duração deste mesmo vídeo precisaria de aproximadamente 19 GB para ser armazenado (AGOSTINI, 2007). Tanto a taxa de transmissão quanto o armazenamento são intratáveis para as tecnologias atuais (AGOSTINI, 2007). Ao mesmo tempo, os vídeos digitais possuem uma forte correlação entre os dados que os compõem, de forma que existe muita redundância na sua representação.

Desta forma, a compressão de vídeo é essencial para as aplicações de vídeo digital. Motivada por esta crescente demanda, a indústria e o meio acadêmico têm investido fortemente no tema de compressão de vídeo, no sentido de buscar elevadas taxas de compressão através de um conjunto de técnicas para eliminação de redundâncias das informações do vídeo. Estas técnicas podem reduzir significativamente a largura de banda necessária para transmissão de vídeos e o tamanho total de memória para seu armazenamento. Por consequência disto, padrões mais atuais de compressão de vídeo podem possibilitar que um vídeo de mais alta resolução seja transmitido com a mesma taxa de transmissão de um vídeo de mais baixa resolução, sem perda de qualidade da imagem.

Com o objetivo de dobrar a taxa de compressão em relação aos padrões existentes até então, o time de especialistas chamado JVT (*Joint Video Team*) (ITU, 2008b) criou o padrão H.264/AVC. O objetivo inicial do JVT foi atingido. O custo disto foi um grande aumento na complexidade computacional dos codecs compatíveis com este padrão. A eficiência em compressão do padrão H.264/AVC em comparação com o padrão mais usado na atualidade, o MPEG-2 (ISO, 2008a), pode ser de até duas vezes, o que representa uma redução de 50% no número de bits do vídeo codificado para a mesma qualidade subjetiva. O preço disto é um aumento na complexidade computacional de aproximadamente quatro vezes, para o decodificador, e de oito vezes, para o codificador, em relação ao padrão MPEG-2 (SUNNA, 2005). Este aumento de complexidade do H.264/AVC somado à necessidade de codificar e decodificar vídeos

de alta-resolução (HDTV) em tempo real impulsiona o meio acadêmico e a indústria a buscar soluções de hardware dedicadas para codificação e decodificação de vídeo. Hoje em dia, existe uma vasta quantidade de trabalhos na literatura tratando de soluções específicas de *hardware* para o codec H.264/AVC, como os trabalhos de Huang et al. (HUANG, 2005a), Lin et al. (LIN, 2007) e Agostini et al. (AGOSTINI, 2007a).

No padrão H.264/AVC foram introduzidas uma série de novas possibilidades de codificação, as quais serão detalhadas no decorrer deste texto. Uma das técnicas, que explora a redundância espacial contida em uma imagem do vídeo, é a predição intra-quadro. Presente no codificador e no decodificador do padrão H.264/AVC, esta predição é gerada para cada macrobloco com base somente nas amostras de uma mesma imagem do vídeo a ser codificado. Esta predição fornece ganhos de compressão quando a predição inter-quadros não encontra um bom resultado ou no caso dos quadros intra, onde a predição inter-quadros não pode ser aplicada (não são permitidas referências a quadros anteriormente codificados). Um vídeo codificado somente usando predição intra-quadro do H.264/AVC pode melhorar a eficiência de codificação se comparado a outros padrões de compressão de imagens estáticas, como o JPEG (ITU, 1992; HUANG, 2005).

A predição intra-quadro representa um aumento na complexidade computacional e latência do codificador H.264/AVC. São definidos 13 modos de predição intra-quadro que devem ser testados para cada macrobloco, o que afeta a complexidade computacional do codificador. A latência no processo de codificação se dá devido à dependência de dados no nível de macroblocos (e de blocos 4x4) da etapa de predição intra-quadro com a etapa que processa os resíduos da predição, formada pelos módulos de transformadas e quantização diretas e inversas. Esta dependência de dados torna impraticável a aplicação de antigas técnicas de *hardware* usadas em codificadores de padrões anteriores de compressão de vídeo, como o *pipeline* entre a etapa de predição e a etapa que processa os resíduos (HUANG, 2005). A complexidade computacional e latência inserida pela predição intra-quadro, afeta o desempenho do codificador H.264/AVC para processar vídeos em tempo-real. A exigência de desempenho é maior quando se torna necessário codificar vídeos de alta definição em tempo-real.

Com isto, a predição intra-quadro em *hardware* para o codificador do padrão H.264/AVC foi tema de vários trabalhos encontrados na literatura, como os de Huang et al. (HUANG, 2005), Suh, Park e Cho (SUH, 2005), Cheng, Ku e Chang (CHENG, 2006), Ku et al. (KU, 2006), Li et al. (LI, 2007), Chang et al. (CHANG, 2007), Sahin e Hamzaoglu (SAHIN, 2007), Kuo e Lin (KUO, 2008) e Lin et al. (LIN, 2009), dentre outros. Até este momento, somente os trabalhos de (KUO, 2008) e (LIN, 2009) são capazes de codificar vídeos de resolução HD 1080p (1920x1088 amostras), conhecido no mercado como Full-HD, a 30 quadros/s.

O presente trabalho tem como objetivo propor e desenvolver uma arquitetura de *hardware* dedicada à predição intra-quadro para o codificador compatível com o padrão H.264/AVC, sendo capaz de codificar vídeos de alta definição em tempo-real, mais precisamente vídeos com resolução HD 1080p a 30 quadros/s. Uma investigação inicial é apresentada com o intuito de explorar o máximo do paralelismo a fim de aumentar a vazão de dados (*throughput*) e assim diminuir o número de ciclos necessários para processar um macrobloco em comparação com os trabalhos de Kuo e Lin (KUO, 2008) e de Lin et al. (LIN, 2009), o que se trata a principal contribuição deste trabalho. Devido à dependência de dados no processo de predição intra-quadro, a redução no número de

ciclos, resultado da exploração do paralelismo, é um caminho para o aumento no desempenho do sistema.

Posteriormente, este trabalho apresenta o desenvolvimento da arquitetura em linguagem de descrição de hardware, mais especificamente, a linguagem VHDL (*VHSIC Hardware Description Language*). Resultados da verificação, síntese e o mapeamento da arquitetura para dispositivos FPGA (*Field Programmable Gate Array*) da Xilinx (XILINX, 2009) são apresentados a seguir. Uma síntese para ASIC (*Application Specific Integrated Circuit*), utilizando a tecnologia *standard-cells* TSMC 0.18 μ m, também é realizada para efeito de comparação com os trabalhos relacionados, a maioria com foco em ASIC. A verificação funcional da arquitetura é feita usando como base um *software* descrito em MATLAB, desenvolvido no escopo deste trabalho, e o software de referência do padrão H.264/AVC, o JM (SUHRING, 2009), desenvolvido pelo JVT. Ao final, uma análise de desempenho é apresentada, considerando o número de ciclos da arquitetura desenvolvida juntamente com os módulos inseridos no laço da codificação intra-quadro (transformadas e quantização diretas e inversas), para avaliar se a arquitetura atinge o desempenho desejado de codificar vídeos HD 1080p a 30 quadros/s.

Este trabalho está inserido no escopo de um projeto do grupo de pesquisa em TV Digital da UFRGS, que tem como objetivo desenvolver um codificador H.264/AVC em *hardware*. Todo desenvolvimento do codificador tem como objetivo o processamento de vídeos de alta definição (HDTV) em tempo-real, mais especificamente resolução HD 1080p a 30 quadros/s. Como resultado deste projeto se espera um protótipo do codificador H.264/AVC em plataforma baseada em dispositivos FPGA, e por esta razão o desenvolvimento da arquitetura tem como foco o mapeamento para estes dispositivos.

O trabalho de Staehler et al. (STAEHLER, 2006), do grupo de pesquisa em TV Digital da UFRGS, já tratou de uma arquitetura de *hardware* específica para a predição intra-quadro, porém, no contexto do decodificador de vídeo H.264/AVC, visando também o processamento para vídeos HDTV. O presente trabalho trata do desenvolvimento da predição intra-quadro no contexto do codificador. A diferença básica é que, no decodificador, somente um modo de predição intra-quadro deve ser calculado para cada macrobloco (no caso da predição Intra 16x16 e de crominância) ou um modo de predição por bloco 4x4 (no caso da predição Intra 4x4), como estiver sinalizado no bitstream. Na predição intra-quadro do codificador, 9 modos de predição para cada bloco 4x4 e 8 modos de predição (4 para Intra 16x16, 4 para crominância) devem ser testados para cada macrobloco, e o codificador deve decidir quais modos e tipo de particionamento deve ser aplicado para cada macrobloco. Desta forma, a predição intra-quadro no codificador é mais complexa e os resultados não podem ser diretamente comparados com os obtidos na implementação do decodificador.

O texto da dissertação foi organizado da seguinte forma. Na seção 2, uma introdução geral sobre o padrão H.264/AVC é apresentada. A seção 3 apresenta uma revisão mais específica sobre a predição intra-quadro, foco deste trabalho, no contexto do codificador compatível com o padrão H.264/AVC. A seção 4 apresenta a arquitetura de *hardware* desenvolvida, seus principais módulos, os resultados de síntese, a metodologia de verificação utilizada e uma análise de desempenho para codificar vídeos em tempo-real. A seção 5 apresenta as comparações dos resultados com alguns trabalhos relacionados encontrados na literatura. A seção 6 conclui o trabalho e indica alguns trabalhos futuros.

2 CONCEITOS DE COMPRESSÃO DE VÍDEO E O PADRÃO H.264/AVC

Neste capítulo, alguns conceitos sobre representação e compressão de vídeo digital serão brevemente introduzidos e uma introdução sobre o padrão de compressão de vídeo H.264/AVC, foco desta dissertação, será apresentada. Uma abordagem mais extensa sobre compressão de vídeo, bem como sobre o padrão H.264/AVC, pode ser encontrada em (RICHARDSON, 2003) e (PURI, 2004). Informações mais detalhadas e precisas sobre o padrão H.264/AVC são apresentadas pela instituição que o definiu (ITU, 2007).

2.1 Captura e Representação do Vídeo Digital

2.1.1 Captura

O vídeo digital é uma aproximação discreta no domínio do tempo e do espaço de uma cena, naturalmente contínua. O vídeo digital é representado por uma sequência de imagens (uma matriz retangular de pontos) capturadas a intervalos regulares no tempo. Cada imagem é capturada por um sensor, usualmente uma matriz bidimensional de CCD (*Charge Coupled Devices*). O CCD é um dispositivo analógico que produz um sinal elétrico proporcional à intensidade luminosa incidente no sensor (RICHARDSON, 2003).

A captura de cada imagem define a amostragem espacial do vídeo. Quanto mais pontos (*pixels*) uma imagem é representada, maior sua resolução. Os *pixels* da imagem são usualmente organizados em formatos pré-definidos. Alguns formatos comuns são o CIF (*Common Intermediate Format*) com 352x288 *pixels*, o Quarter CIF, ou QCIF (*Quarter Common Intermediate Format*), com 176x144 *pixels* e o VGA (*Video Graphics Array*) com 640x480 *pixels*.

O intervalo com que é feita cada captura de imagem define a amostragem temporal do vídeo. Com uma taxa de captura suficientemente alta em relação às alterações de uma cena, é possível preservar a noção de movimento. Quanto maior a taxa de captura, mais suave é o movimento da cena de um quadro para o outro. Porém, neste caso, mais amostras devem ser capturadas e armazenadas. A escolha da taxa é feita de forma que o olho humano não perceba a transição entre as imagens do vídeo. Taxas de captura usualmente utilizadas estão na faixa de 15 a 60 imagens capturadas por segundo (RICHARDSON, 2003).

2.1.2 Espaços de cores e sub-amostragem de cores

A partir da fisiologia do sistema visual humano e da teoria das cores pode-se representar cada *pixel* de uma imagem do vídeo por três componentes (ou amostras) de

cor. O espaço de cores mais comum e conhecido para representar imagens digitais é o RGB (*Red, Green, Blue*). O RGB utiliza três matrizes distintas para representar as três cores primárias captadas pelo sistema visual humano: vermelho, verde e azul.

Por outro lado, o espaço de cores mais usado para compressão de vídeo é o YCbCr (luminância, croma azul e croma vermelho, do inglês *Luminance, Chrominance Blue, Chrominance Red*). Este espaço de cores é usado para compressão porque, ao contrário do RGB, a informação de cor (croma) está completamente separada da informação de brilho (luminância). Neste caso podem ser aplicadas ferramentas de compressão distintas para luminância e croma.

Outra vantagem do YCbCr é que pode ser aplicada uma sub-amostragem somente na informação de croma. Como o sistema visual humano é mais sensível ao brilho do que à croma, esta sub-amostragem não representa impacto visual na imagem e, ao mesmo tempo, reduz significativamente os dados para representação do vídeo. Na sub-amostragem 4:2:0, a mais usada no H.264/AVC, para cada 4 amostras de luminância estão associadas uma amostra de Cb e uma amostra de Cr, ou seja, somente metade da amostragem de cada componente de cor em cada direção, horizontal e vertical, é utilizada. Outro tipo de sub-amostragem é o 4:2:2, na qual os componentes de cor sofrem amostragem somente na direção horizontal, mantendo a amostragem na direção vertical. No caso do 4:2:0, somente com a sub-amostragem já existe um ganho de compressão de 50% em relação à 4:4:4, onde não existe sub-amostragem de cor (RICHARDSON, 2003). Esta técnica se baseia na limitação do olho humano para eliminar parte da informação irrelevante presente nas imagens.

2.1.3 Métricas para Avaliação de Qualidade

A qualidade de um vídeo digital é um parâmetro complexo para definir e para avaliar. Os critérios subjetivos são muito difíceis de medir. Existem alguns critérios objetivos para comparação de imagens (RICHARDSON, 2003), que se baseiam na comparação dos *pixels* da imagem original com os *pixels* da imagem após a decodificação. Esta técnica é útil no caso presente quando se quer determinar a fidelidade da reconstrução de uma imagem. O critério mais conhecido e aceito pela comunidade é o PSNR (*Peak Signal-to-Noise Ratio*) (GHANBARI, 2003), definido pela equação (1).

$$PSNR_{dB} = 20 \cdot \log_{10} \left(\frac{MAX}{\sqrt{MSE}} \right) \quad (1)$$

Em (1), MAX é o máximo valor de representação de uma amostra ($2^n - 1$, sendo n o número de bits para representar cada amostra) e MSE é o erro médio quadrático (*Mean-Squared Error*), sendo definido em (2), onde mn é a dimensão da imagem em *pixels*, O e R são as imagens original e reconstruída, respectivamente.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (R_{i,j} - O_{i,j})^2 \quad (2)$$

O SAD (*Sum of Absolute Differences*) também é usado como critério para avaliar a distorção entre duas imagens, feito a partir do somatório das diferenças absolutas para cada ponto do quadro original e do quadro reconstruído (RICHARDSON, 2003). A função de cálculo do SAD é mostrada em (3), onde m e n são as dimensões do quadro a ser comparado e O e R são as amostras dos quadros original e reconstruído,

respectivamente. Este critério é muito usado por sua baixa complexidade computacional em relação ao MSE (RICHARDSON, 2003).

$$SAD = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |R_{i,j} - O_{i,j}| \quad (3)$$

Estas técnicas são úteis para avaliar a qualidade da reconstrução de uma imagem após a compressão com perdas, como será detalhado nas próximas seções.

2.2 Conceitos Básicos de Compressão de Vídeo

Existem dois tipos de técnicas para compressão de dados em geral: sem perdas de informação (*lossless compression*) ou com perdas de informação (*lossy compression*) (RICHARDSON, 2003). Técnicas de compressão sem perdas de informação codificam os dados de forma que possam ser reconstruídos de forma idêntica aos dados originais. Este tipo de compressão é muito usado para codificação de arquivos onde não é admitida a perda de informação. Tais técnicas podem atingir baixas taxas de compressão, de 3 a 4 vezes usando JPEG-LS, por exemplo (RICHARDSON, 2003).

2.2.1 Técnicas de Compressão com Perdas de Informação

Técnicas de compressão com perdas de informação codificam os dados, sendo que a reconstrução resulta em alguns dados diferentes dos dados originais. É possível atingir taxas de compressão muito maiores. Estas técnicas são extensivamente usadas em padrões de compressão de imagens e vídeos. As técnicas para compressão de imagens e vídeos exploram as limitações do olho humano para selecionar a informação, de forma que a reconstrução da imagem tenha uma diferença visual imperceptível ao olho humano em comparação com a imagem original.

A própria representação do vídeo utilizando sub-amostragem de cores (seção 2.1.2) é um exemplo de técnica de compressão com perdas de informação. Outra técnica que insere perdas na compressão é a quantização. Ela atua no domínio das frequências, ou seja, a imagem primeiramente deve ser transformada para este domínio. A quantização elimina as frequências de menor amplitude contidas em uma imagem, as quais são imperceptíveis ao olho humano. Na seção 2.3, a quantização no padrão H.264/AVC será detalhada.

2.2.2 Redundâncias de Dados na Representação de Vídeos

Além do ganho da compressão com perdas de informação, os vídeos digitais propiciam uma série de outras técnicas de compressão devido ao elevado grau de informações redundantes na sua representação. Existem três tipos principais de redundância: espacial, temporal e entrópica (ou estatística):

- Redundância Espacial: também chamada de redundância intra-quadro, representa a correlação existente entre os *pixels* espacialmente distribuídos de um quadro. Esta redundância é percebida tanto no domínio espacial quanto no domínio das frequências (AGOSTINI, 2007);
- Redundância Temporal: também chamada de redundância inter-quadro, representa a correlação existente entre os *pixels* dos quadros temporalmente próximos em um vídeo. Esta redundância aumenta quando o vídeo tem pouco

movimento ou quanto mais alta for a taxa de amostragem temporal (número de imagens amostradas por segundo);

- Redundância Entrópica: está relacionada com as probabilidades de ocorrência dos símbolos codificados. Há maneiras de representar os símbolos codificados do vídeo com menor número de bits, reduzindo o volume de informações.

Para reduzir os demais tipos de redundância, padrões de compressão de vídeo atuais utilizam largamente o conceito de codificação preditiva (RICHARDSON, 2003). Explorando a correlação existente entre dois dados do vídeo, um dado é enviado e o dado seguinte é aproximado, ou seja, calculado com base no anterior, sendo que somente o resíduo (diferença entre o dado original e sua predição) é enviado ao decodificador (RICHARDSON, 2003). O decodificador reconstrói o dado predito, com base no dado original, sem perda de informação.

A predição intra-quadro do padrão H.264/AVC, foco desta dissertação, realiza predição sobre as amostras espacialmente distribuídas do quadro, enviando somente o resíduo da predição. A predição inter-quadro, mais conhecida como o par estimação/compensação de movimento, explora a redundância temporal, fazendo predição da imagem a ser codificada com base em imagens temporalmente próximas. Por fim, a codificação de entropia atua na representação dos dados do vídeo para reduzir sua redundância estatística.

Os padrões de compressão de vídeo combinam todas estas técnicas anteriormente citadas para atingir elevadas taxas de compressão. A seção 2.3 irá mostrar como cada uma destas etapas compõe um codificador/decodificador e detalhar cada uma delas no contexto do padrão H.264/AVC, foco desta dissertação.

2.3 Introdução ao padrão H.264/AVC

2.3.1 Histórico do Padrão H.264/AVC

O H.264/AVC é o mais novo padrão de compressão de vídeo, especificado pelo JVT (*Joint Video Team*), formado por um grupo de especialistas do VCEG (*Video Coding Experts Group*) da ITU (*International Telecommunication Union*) (ITU, 2008) e do MPEG (*Moving Picture Experts Group*) da ISO/IEC (*International Organization for Standardization/International Electrotechnical Commission*) (ISO, 2008). O nome do padrão adotado pela ITU foi H.264, como o sucessor do padrão H.263 (ITU, 2000) e pela ISO/IEC foi MPEG-4 parte 10 – AVC (*Advanced Video Coding*), como uma parte do desenvolvimento do padrão MPEG-4 (ISO, 2008). O nome H.264/AVC foi definido para se referir ao padrão de forma equilibrada entre os nomes dados pela ITU e pela ISO. O padrão é referenciado por este nome na maioria das publicações da área e será usado no decorrer do texto.

A primeira versão do padrão H.264/AVC foi aprovada em outubro de 2003 (ITU, 2003), e define o processo de decodificação de vídeo e o formato do *bitstream*. Seu processo de codificação não é normatizado, sendo que o projeto de construção de um codificador é mais livre para suportar ou não as ferramentas definidas no padrão. O único requisito do codificador é gerar um *bitstream* de saída que seja válido para decodificação em um decodificador definido conforme o padrão. Em 2004, uma nova versão do padrão, que prevê aplicações de vídeo com qualidade profissional, foi proposta. Esta versão é conhecida como *Fidelity Range Extensions* (FRExt) (ITU, 2004)

e foi anexada como uma extensão do padrão H.264/AVC em 2005 (ITU, 2005). Em novembro de 2007, o grupo JVT aprovou o padrão SVC (*Scalable Video Coding*), que dá suporte a vídeos escaláveis, e foi publicado como uma extensão do H.264/AVC (ITU, 2007).

2.3.2 Estrutura de Codificação

A codificação do vídeo é feita para cada imagem que o compõe. Cada imagem codificada pode ser um quadro, quando o vídeo é progressivo ou entrelaçado, ou um campo (*field*), quando o vídeo é entrelaçado (RICHARDSON, 2003). No vídeo entrelaçado a imagem é particionada em linhas pares e ímpares de amostras, como utilizado nas transmissões de vídeo analógico. Este trabalho trata somente o tipo quadro (vídeo progressivo), portanto, os termos “quadro” e “imagem” representam a mesma estrutura. A cada quadro é associado um número que define sua ordem de exibição, o *Picture Order Count* (POC). Esta ordem não é necessariamente igual à ordem de codificação dos quadros, pois os quadros podem, em certas situações, ser codificados fora da ordem de exibição. Os quadros codificados anteriormente, organizados em duas listas (0 e 1), podem ser utilizados como quadros de referência para a predição de outros quadros.

Cada quadro pode ser codificado em um *slice* ou ser particionado em vários *slices*. Um *slice* é um conjunto inteiro de macroblocos ordenados consecutivamente na ordem *raster* (da esquerda para direita, de cima para baixo) dentro de uma imagem (ITU, 2007). O *slice* representa uma entidade auto-contida para fins de codificação (PURI, 2004). Neste trabalho, por simplificação, cada quadro será codificado utilizando somente um *slice*. O padrão define cinco tipos de *slice*: I (intra), P (preditivo), B (bi-preditivo), SI (*Switch-I*) ou SP (*Switch-P*).

Cada *slice* é formado por um conjunto de macroblocos, onde cada macrobloco contém 16x16 amostras de luminância (Y) e duas componentes de croma associadas (Cb e Cr). Usando a sub-amostragem de cor 4:2:0, a mais utilizada no padrão, cada componente de croma é formado por um bloco de 8x8 amostras. Cada amostra, seja de luminância ou croma, possui 8 bits. O FRExt (ITU, 2004) estendeu o padrão para poder utilizar sub-amostragens 4:2:2 e 4:4:4 e até 12 bits por amostra (ITU, 2005). Os macroblocos são classificados em três tipos: I, P e B. Um *slice* do tipo I (Intra) pode conter somente macroblocos do tipo I. Um *slice* do tipo P pode conter macroblocos do tipo P e I e um *slice* do tipo B pode conter macroblocos do tipo B e I. Os *slices* SI e SP são análogos aos *slices* I e P, porém usados em situações específicas de troca de contexto para aplicações de *streaming* de vídeo, e não serão tratados nesta dissertação.

Os macroblocos do tipo I são codificados usando codificação intra-quadro, que somente requer amostras de um mesmo *slice* para codificação. O padrão define dois tipos de predição para amostras de luminância: Intra 16x16 e Intra 4x4, com 4 e 9 modos de predição, respectivamente. Quatro modos idênticos ao Intra 16x16 são aplicados a blocos de croma. Um quadro contendo somente *slices* do tipo I (formado por macroblocos I) é o primeiro a ocorrer na sequência de decodificação. Existe um tipo especial de quadro, denominado IDR (*Instantaneous Decoder Refresh*), que zera as listas 0 e 1 de referência, ou seja, todos os quadros posteriores a ele não podem fazer referências a quadros anteriores a ele, na ordem de exibição. Por este motivo, este é o primeiro quadro a ser decodificado. Por se tratar do foco principal desta

dissertação, o capítulo 3 é dedicado especificamente aos conceitos da predição intra-quadro.

Os macroblocos do tipo P são codificados usando a predição inter-quadros, a partir de quadros de referência previamente processados pelo codificador. Um macrobloco codificado no modo inter-quadros pode ser dividido em partições de macroblocos, isto é, em blocos de 16x16, 16x8, 8x16 ou 8x8 amostras de luminância. Se o tamanho de partição escolhido for o 8x8, então cada partição 8x8 é chamada de sub-macrobloco, podendo ser dividida novamente em partições de tamanho 8x4, 4x8 ou 4x4 amostras. Cada partição de no mínimo 8x8 amostras pode utilizar como referência uma partição contida em um quadro da lista 0. Caso esta partição seja subdividida em partições menores, as partições menores só poderão referenciar partições do mesmo quadro referenciado na partição 8x8.

Os macroblocos do tipo B também são codificados usando a codificação inter-quadros. Cada partição de macrobloco com pelo menos 8x8 amostras de luminância pode utilizar como referência uma partição correspondente em um ou dois quadros diferentes, um na lista 0 e outro na lista 1. Assim como nos macroblocos P, caso esta partição for subdividida em partições menores, as partições menores só poderão referenciar partições dentro dos mesmos quadros referenciados na partição 8x8.

Além dos macroblocos apresentados acima, o padrão H.264/AVC apresenta um tipo de macrobloco chamando *skip*. O macrobloco *skip* é um tipo especial de macrobloco no qual nenhuma informação é colocada no *bitstream*, exceto a sinalização de que este macrobloco é *skip*. O decodificador calcula um vetor de movimento para o macrobloco *skip* e reconstrói o macrobloco usando compensação de movimento usando o primeiro quadro de referência da lista 0 (RICHARDSON, 2003).

2.3.3 Perfis e Níveis

A primeira versão do padrão H.264/AVC (ITU, 2003) definiu três diferentes perfis: *Baseline*, *Main* e *Extended*. Cada perfil suporta um grupo particular de funções de codificação e especifica o que é necessário para cada codificador e decodificador que seguem este perfil.

O perfil *Baseline* é direcionado a aplicações como videotelefonia, videoconferência e vídeo sem fio. O perfil *Baseline* suporta codificação intra e inter (usando somente *slices* I e P) e uma codificação de entropia com códigos de comprimento variável adaptativos ao contexto (CAVLC). O perfil *Main* é focado na transmissão de televisão e no armazenamento de vídeo. O perfil *Main* inclui o suporte para vídeo entrelaçado, o suporte à predição inter-quadros utilizando *slices* do tipo B e utilizando predição ponderada (média ponderada entre as predições inter-quadros de dois macroblocos localizados em dois quadros de referência distintos) e o suporte à codificação de entropia utilizando codificação aritmética adaptativa ao contexto (CABAC). O perfil *Extended* é mais voltado para aplicações em *streaming* de vídeo e não suporta vídeo entrelaçado e CABAC, mas agrega modos para habilitar uma troca eficiente entre *bitstreams* codificados (através de *slices* do tipo SP e SI) e melhora a robustez a erros de transmissão (através do particionamento de dados).

Para os três perfis definidos na primeira versão do padrão, a relação entre os elementos de cores é fixa. Esta relação é de 4:2:0 para os elementos Y, Cb e Cr. Os perfis *Baseline*, *Main* e *Extended* também possuem outra característica em comum,

neste caso, relacionada à quantidade de bits utilizados por amostra. Nos três perfis, são usados 8 bits por amostra de luminância e croma.

Estes três perfis inicialmente propostos pelo padrão H.264/AVC foram focados em vídeos de entretenimento ou de qualidade. Mas estes perfis não incluíram suporte para vídeos com resoluções mais elevadas, como as necessárias em ambientes profissionais. Para responder às exigências deste tipo de aplicação, uma continuação do projeto JVT foi realizada para adicionar novas extensões para as capacidades do padrão original. Estas extensões foram chamadas de extensões para alcance de fidelidade (*Fidelity Range Extensions, ou FRExt*). O FRExt produziu um grupo de quatro novos perfis chamados coletivamente de perfis *High* (SULLIVAN, 2004).

Os perfis *High* possuem algumas características comuns que são inovadoras em relação aos perfis originais: suportam um tamanho de bloco adaptativo para a transformada (4x4 ou 8x8), matrizes de quantização baseadas em percepção e uma representação sem perdas de regiões específicas do conteúdo do vídeo (SULLIVAN, 2004).

O perfil *High* inclui vídeo com 8 bits por amostra e com relação de cor 4:2:0. O perfil *High 10* suporta todas ferramentas do perfil *High* e ainda vídeos com 8, 9 e 10 bits por amostra, também com uma relação de cor 4:2:0. O perfil *High 4:2:2* suporta todas ferramentas do *High 10* e inclui ainda suporte à sub-amostragem de cor 4:2:2 (ITU, 2007). As ferramentas suportadas pelos diferentes perfis estão resumidas na Figura 2.1.

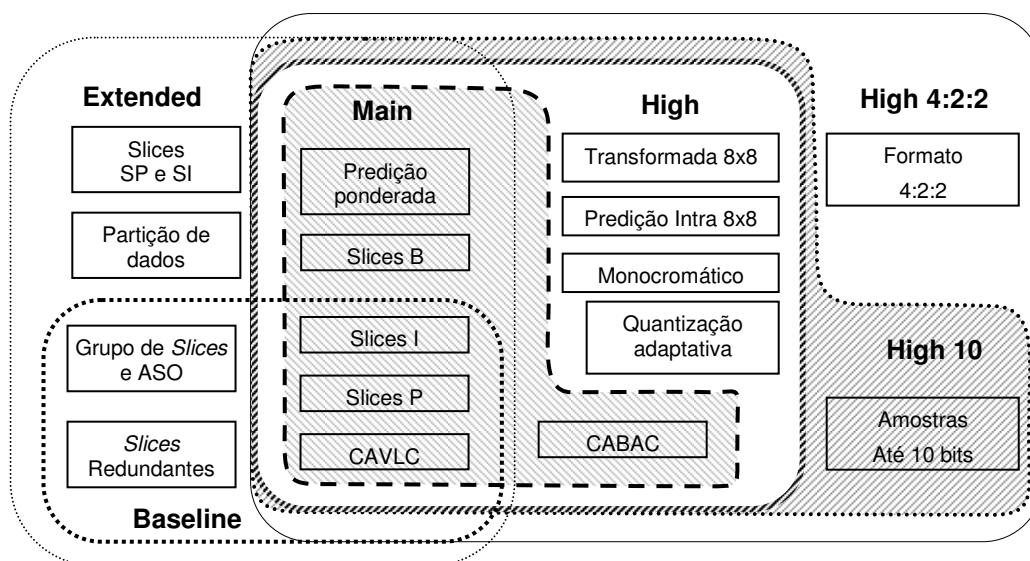


Figura 2.1: Perfis do H.264/AVC (ROSA, 2009).

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 do codificador ou decodificador, como mostra a Tabela 2.1. Com a definição do nível utilizado, é possível deduzir o número máximo de quadros de referência e a máxima taxa de bits que podem ser utilizados (SULLIVAN, 2004).

Tabela 2.1: Níveis do H.264/AVC

1	99	1 485	64
1b	99	1 485	128
1.1	396	3 000	192
1.2	396	6 000	384
1.3	396	11 880	768
2	396	11 880	2 000
2.1	792	19 800	4 000
2.2	1 620	20 250	4 000
3	1 620	40 500	10 000
3.1	3 600	108 000	14 000
3.2	5 120	216 000	20 000
4	8 192	245 760	20 000
4.1	8 192	245 760	50 000
4.2	8 704	522 240	50 000
5	22 080	589 824	135 000
5.1	36 864	983 040	240 000

Fonte: (ITU, 2007), p. 283.

2.3.4 Formato do Vídeo Codificado

Uma interessante inovação do padrão H.264/AVC é que ele faz uma clara distinção entre a organização do vídeo codificado e sua transmissão via rede. O padrão classifica a informação do vídeo em dois diferentes níveis de abstração, chamados de camada de vídeo codificado (VCL – *Vídeo Coding Layer*) e camada de abstração de rede (NAL – *Network Abstraction Layer*) (RICHARDSON, 2003).

Os dados de saída do processo de codificação de vídeo estão na camada VCL, sendo formados por uma sequência de bits que representam os dados do vídeo codificado. Estes dados são mapeados para unidades NAL antes da transmissão ou armazenamento. Uma sequência de vídeo codificado é representada por uma sequência de unidades NAL que podem ser transmitidas sobre uma rede baseada em pacotes, via um link de transmissão de *bitstream* ou ainda armazenados em um arquivo. A razão para a existência das duas camadas é aumentar a flexibilidade do transporte dos dados de vídeo (VCL) sobre diversos protocolos de rede, abrangendo uma diversidade de aplicações (PURI, 2004).

2.4 Núcleo do Codec H.264/AVC

O núcleo do codec compreende os principais blocos de um codificador e de um decodificador, bem como suas funcionalidades. Todas as técnicas a seguir são aplicadas para cada macrobloco no sentido de explorar as redundâncias do vídeo para obter a melhor codificação. Um codificador H.264/AVC é composto pelos blocos de predição intra-quadro, predição inter-quadros, transformadas diretas e inversas (T e TI), quantização direta e inversa (Q e QI), filtro redutor do efeito de bloco e codificação de entropia, como apresentados na Figura 2.2.

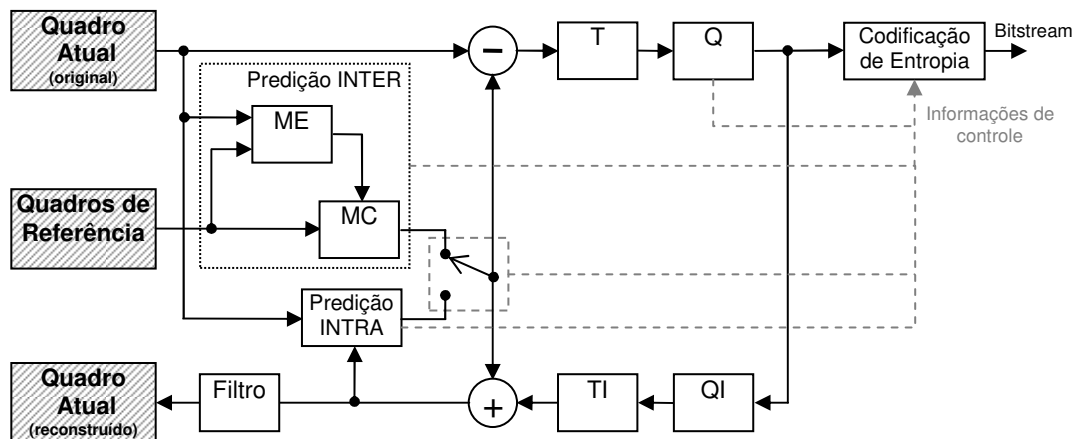


Figura 2.2: Diagrama em blocos de um codificador H.264/AVC (Modificada a partir de (AGOSTINI, 2007)).

Em resumo, o funcionamento de um codificador de vídeo H.264/AVC é o seguinte: cada macrobloco do quadro atual (original), a ser codificado, é subtraído de um macrobloco predito usando predição intra, ou de um macrobloco contido em um quadro de referência. No caso do primeiro macrobloco a ser codificado, um tipo especial de predição intra-quadro é aplicado, onde todas as amostras assumem o valor 128 (exatamente a metade da faixa de valores para amostras de 8 bits, ou seja, de 0 a 255). O que resulta da subtração entre os macroblocos original e predito é um macrobloco de resíduos (diferença *pixel a pixel* entre o macrobloco atual e o macrobloco predito) que é particionado em blocos 4x4 e processado pelos blocos de transformada (bloco T) e quantização (bloco Q), gerando os coeficientes quantizados. Os coeficientes quantizados, bem como as demais informações de controle suficientes para decodificação (indicadores de modo de predição e vetores de movimento, por exemplo) são enviados para a etapa de codificação de entropia, que explora a redundância estatística destas informações e gera uma sequência de bits que forma o macrobloco codificado. O conjunto de todos os macroblocos codificados e organizados em diferentes quadros forma o vídeo codificado. A sequência de bits que representa o vídeo codificado é denominada *bitstream*.

Como a etapa de quantização gera perdas, o quadro reconstruído no decodificador não será igual ao quadro original. A fim de evitar diferenças no cálculo dos resíduos no codificador e no decodificador é necessário que o codificador também realize a reconstrução do quadro. Alguns quadros reconstruídos, escolhidos pelo codificador, poderão servir de referência para predição dos próximos quadros. Deste modo, os coeficientes quantizados (saída do bloco Q, na Figura 2.2) são enviados também a etapa de reconstrução, composta pelas transformadas e quantização inversa (TI e QI). Estes resíduos, já com perdas, são somados à predição para servirem de referência para a predição de outro quadro, no caso da predição inter, ou de outro bloco no mesmo quadro, no caso da predição intra. Isto insere um custo alto no processo de codificação, especialmente na predição Intra, que será detalhada no capítulo 3 deste texto. O filtro redutor de efeitos de bloco, opcional em outros padrões, mas normatizado no H.264/AVC, é usado no caminho para reconstrução do quadro de referência para a predição inter. O bloco de predição inter-quadros, apresentado na Figura 2.2, é formado pelo par estimação/compensação de movimento (ME, do inglês, *Motion Estimation* e

MC, *Motion Compensation*), os quais representam a maior complexidade computacional do codificador.

A Figura 2.4 apresenta o diagrama em blocos de um decodificador H.264/AVC, que é composto pelos blocos de predição intra-quadro, predição inter-quadros, transformadas e quantização inversas (TI e QI), filtro redutor de efeitos de bloco e decodificação de entropia.

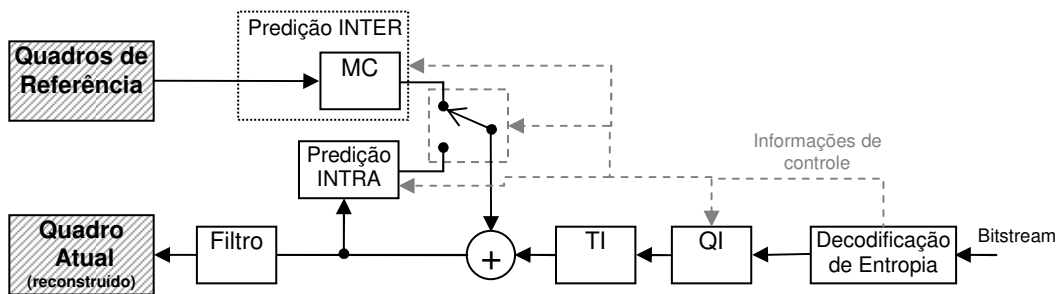


Figura 2.4: Diagrama em blocos de um decodificador H.264/AVC (Modificada a partir de (AGOSTINI, 2007)).

O funcionamento de um decodificador de vídeo H.264/AVC é similar ao caminho de reconstrução do codificador. O *bitstream* de entrada é decodificado pelo decodificador de entropia, sendo que os resíduos quantizados são processados pelos blocos QI (quantização inversa) e TI (transformadas inversas). No *bitstream* também estão contidas informações de controle essenciais para predição e reconstrução do vídeo, tais como o tipo do macrobloco, modos de predição e vetores de movimento. A predição sinalizada no *bitstream* é realizada pelos blocos MC e INTRA, sendo que o macrobloco predito é somado com os resíduos resultantes da etapa TI. A imagem reconstruída é posteriormente filtrada para ser exibida e armazenada para futuras referências.

Os projetos do codificador e do decodificador podem ser semelhantes neste nível de abstração, mas são essencialmente diferentes. Primeiramente, o decodificador de vídeo, por ser normatizado, deve ser totalmente compatível com o padrão (em algum perfil/nível, como mostrado na seção 2.2.4), ou seja, todas as técnicas de compressão devem ser implementadas. Já no codificador podem ser decididas quais as técnicas serão ou não incluídas no projeto, desde que o codificador produza um *bitstream* compatível na saída, ou seja, que um decodificador compatível com o padrão consiga decodificá-lo. Desta forma, o projeto do codificador possui muito mais flexibilidade que o projeto do decodificador.

Apesar disto, para atingir um nível aceitável de qualidade da imagem codificada e eficiência de compressão, o codificador deve escolher em tempo de processamento entre as inúmeras formas de codificação que o H.264/AVC prevê para cada macrobloco, por exemplo, o tipo de predição (intra-quadro ou inter-quadros), o tamanho de bloco usado nas predições intra-quadro ou inter-quadros, se o modo *skip* será usado ou não, se será usada bi-predição, etc. Esta decisão é crucial e deve ser bem tomada, caso contrário os benefícios proporcionados pelo H.264/AVC podem ser reduzidos (PURI, 2004).

A decisão do modo de predição, apesar de não normatizada, deve ser feita baseada em técnicas de otimização taxa-distorção (WIEGAND, 2003; SULLIVAN, 1998). A otimização taxa-distorção é o processo que relaciona a taxa de bits de um vídeo

codificado com o grau de distorção deste vídeo quando decodificado, após a codificação com perdas. Em (4), \mathbf{D} representa o grau de distorção, \mathbf{R} representa a taxa de bits do vídeo codificado e \mathbf{J} é o custo que deve ser minimizado.

$$J = D + \lambda \cdot R \quad (4)$$

Minimizar o custo \mathbf{J} significa encontrar um compromisso entre a taxa (número de bits usados para representar o vídeo codificado) e a qualidade (expressa em PSNR). O parâmetro λ é um multiplicador de Lagrange, que relaciona distorção e taxa de bits, e influencia diretamente neste compromisso entre taxa e distorção. Na prática, este parâmetro é obtido experimentalmente como função do parâmetro de quantização (PURI, 2004).

Tomar uma decisão ótima do modo de predição envolve testar cada combinação de modos de predição dentro de um macrobloco, gerando um custo \mathbf{J} . Para gerar o custo \mathbf{J} de forma ótima é preciso medir de forma exata os parâmetros \mathbf{D} e \mathbf{R} . Porém, diante da enorme combinação de possibilidades de codificação e do limite computacional de um sistema para determinar este custo em um tempo razoável, na prática o custo \mathbf{J} é medido de forma aproximada. O parâmetro \mathbf{D} é usualmente medido usando um critério de similaridade. O MSE fornece uma boa aproximação do critério de similaridade mais utilizado, o PSNR. A taxa de bits é calculada pelo número de bits usado para codificar as informações de controle (modos de predição, tipos de particionamento, etc.) e resíduos após a codificação de entropia. Isto envolve codificar o macrobloco diversas vezes, avaliando todos os modos de predição, antes de decidir quais técnicas de codificação serão usadas e assim codificar o macrobloco com as técnicas escolhidas.

Para aplicações de codificação de vídeos em tempo-real, foco deste trabalho, o custo computacional para determinar o custo \mathbf{J} para cada macrobloco é crítico. A complexidade computacional para determinar o custo \mathbf{J} pode tornar o processo de codificação tão complexo de forma a inviabilizar a codificação de vídeos em tempo-real. O custo computacional cresce com o aumento da resolução. No caso de vídeos de alta resolução, algumas aproximações devem ser adotadas, como usar o SAD para medir o critério de distorção e aproximar a taxa de bits final do vídeo codificado. Em resumo, a complexidade do processo de escolha do modo de predição utilizado pode exigir recursos computacionais além dos disponíveis. É necessário um compromisso de projeto entre o custo do projeto e o desempenho do codificador.

Cada um dos blocos do codificador e do decodificador será brevemente detalhado nas próximas seções, de acordo com sua funcionalidade no H.264/AVC. Por se tratar do principal foco desta dissertação, o bloco de predição intra-quadro será abordado no capítulo 3, com mais detalhes.

2.4.1 Estimação de Movimento (ME)

A Estimação de Movimento (ME), presente apenas no codificador H.264/AVC, forma, juntamente com a Compensação de Movimento (MC), a predição inter-quadros, como mostrado na Figura 2.2. A predição inter-quadros é a responsável pela maior complexidade computacional de um codificador H.264/AVC (PURI, 2004). Este elevado custo computacional é devido às diversas ferramentas inovadoras inseridas nesta etapa. Devido a estas diversas inovações, estes blocos fornecem o maior potencial de ganhos, em termos de compressão, em relação aos padrões anteriores de compressão de vídeos (WIEGAND, 2003, RICHARDSON, 2003).

A Figura 2.4 mostra o processo de estimação de movimento. Este processo tem o objetivo de localizar, nos quadros de referência, qual o bloco mais semelhante ao bloco do quadro atual. A busca geralmente é feita em uma área ao redor do bloco em questão, denominada área de pesquisa. Assim que o melhor casamento é encontrado, a ME deve gerar um vetor de movimento indicando o deslocamento da posição deste bloco do quadro atual no quadro de referência. O valor do vetor de movimento é enviado junto à codificação do macrobloco no *bitstream*.

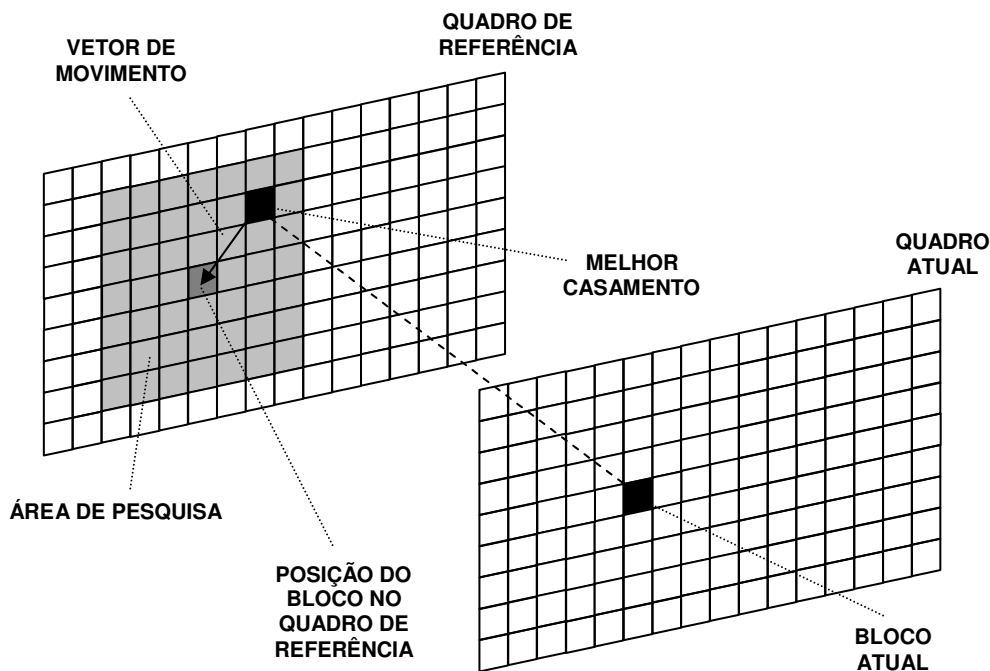


Figura 2.4: Determinação do vetor de movimento para um bloco (Modificada a partir de (PORTO, 2008)).

Por simplificação, a estimação de movimento é aplicada apenas à componente de luminância do macrobloco. Como as componentes de crominância possuem a metade da resolução horizontal e vertical da componente de luminância, então basta uma divisão por dois nas componentes horizontal e vertical de cada vetor de movimento de luminância para serem aplicadas aos blocos de crominância.

A principal inovação do H.264/AVC com relação a ME e MC é a utilização de tamanhos de blocos variáveis. Ao invés da estimação de movimento gerar um vetor de movimento para cada macrobloco, o padrão H.264/AVC permite o uso de partições de macrobloco e partições de sub-macroblocos, sendo que cada partição pode ter seu próprio vetor de movimento. As partições de macroblocos possuem tamanhos de 16x16, 8x16, 16x8 e 8x8 (RICHARDSON, 2003), como está apresentado na Figura 2.5. Quando o tipo de particionamento escolhido é 8x8, cada partição (denominada sub-macrobloco) ainda pode ser dividida em mais quatro tipos (8x8, 8x4, 4x8 e 4x4), como mostrado na Figura 2.6.

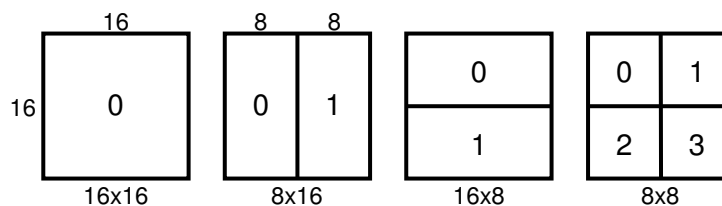


Figura 2.5: Divisão do macrobloco em partições de macrobloco (AGOSTINI, 2007).

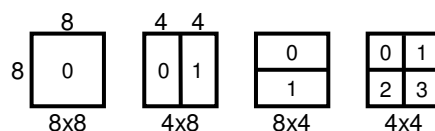


Figura 2.6: Divisão do sub-macrobloco em partições (AGOSTINI, 2007).

Maiores detalhes sobre a estimação de movimento não serão abordados neste texto, podendo ser encontrados em (AGOSTINI, 2007; PORTO, 2008).

2.4.2 Compensação de Movimento (MC)

A compensação de movimento (Figuras 2.2 e 2.3), presente no codificador e no decodificador do padrão H.264/AVC, é o processo inverso da estimação de movimento, e deve ser compatível com as operações feitas naquela etapa. Dados os vetores de movimento e os índices dos quadros de referência, a compensação de movimento localiza no quadro de referência o bloco que foi escolhido na etapa de estimação de movimento para reconstruir o bloco do quadro atual.

A MC no codificador pode ser simplificada, para se adaptar às simplificações possíveis feitas da ME. No entanto, a MC do decodificador deve ser compatível com todas as ferramentas previstas em um perfil do padrão H.264/AVC, para que seja possível decodificar o vídeo gerado por diversos codificadores compatíveis com este mesmo perfil.

As inovações do padrão H.264/AVC com relação ao par ME/MC são as seguintes:

- Múltiplos tamanhos de partições de macroblocos, sendo que cada partição pode ter seu próprio vetor de movimento;
- Múltiplos quadros de referência anteriores e posteriores ao quadro atual na ordem de exibição;
- Predição dos vetores de movimento para codificação de vetores de movimento diferenciais;
- Tratar vetores de movimento que apontam para fora dos limites do quadro;
- Precisão de $\frac{1}{4}$ de pixel para os vetores de movimento de luminância;
- Predições bi-preditiva, ponderada e direta para *slices* do tipo B;
- Macroblocos do tipo *skip* para *slices* dos tipos P e B.

Maiores detalhes sobre a compensação de movimento não serão abordados neste texto, podendo ser encontrados em (AZEVEDO, 2006).

2.4.3 Transformadas Diretas (T)

O bloco T da Figura 2.2 é responsável pelas transformadas diretas dos blocos de resíduo de predição e está presente apenas no codificador. O padrão H.264/AVC define três tipos de transformadas para blocos de resíduo da predição:

- Transformada Hadamard para blocos de 4x4 formados por coeficientes DC de luminância quando o macrobloco é predito usando o modo Intra 16x16;
- Transformada Hadamard para blocos 2x2 formados por coeficientes DC de crominância
- Transformada inteira baseada na DCT 2-D (*Discrete Cosine Transform*) para os demais blocos 4x4.

O bloco T separa as 16x16 amostras de luminância (Y) e 8x8 amostras de crominância (Cb e Cr) para as transformadas, como mostra a Figura 2.7.

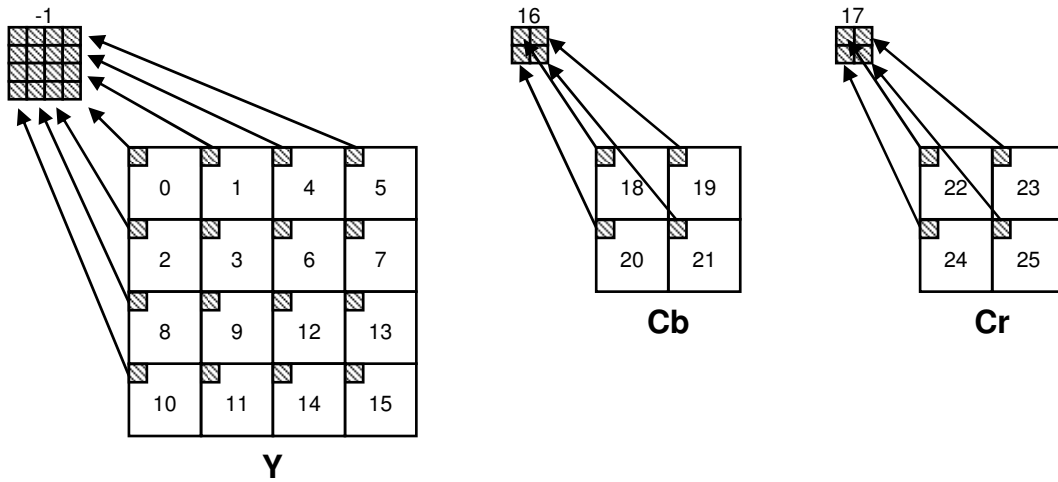


Figura 2.7: Ordem de processamento das amostras no bloco T (AGOSTINI, 2007).

Ainda na Figura 2.7 é ilustrado como os blocos 4x4 (Y) e 2x2 (Cb e Cr) de resíduos DC são formados. Além disto, a numeração dos blocos representa a ordem com que são processados os blocos pela transformada direta. O bloco “-1” na Figura 2.7 só é formado quando o tipo de predição para o macrobloco é Intra 16x16.

A transformada baseada na DCT 2-D (FDCT 2-D), definida em (5), é aplicada aos blocos 0 a 15 (Y) e aos blocos 18 a 25 (Cb e Cr) mostrados na Figura 2.7.

$$Y = C_f X C_f^T \otimes E_f = \begin{pmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} X \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 2 & -1 \end{bmatrix} \end{pmatrix} \otimes \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ \frac{ab}{2} & \frac{ab}{4} & \frac{ab}{2} & \frac{ab}{4} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix} \quad (5)$$

$$a = \frac{1}{2}, \quad b = \sqrt{\frac{2}{5}} \quad (6)$$

Em (5), \mathbf{X} é a matriz 4x4 de entrada, \mathbf{C}_f é a matriz da FDCT 1-D inteira, \mathbf{C}_f^T é a transposta da matriz da FDCT e \mathbf{E}_f é a matriz de fatores de escala. O símbolo \otimes na equação indica uma multiplicação escalar. As letras \mathbf{a} e \mathbf{b} na matriz \mathbf{E}_f são constantes definidas em (6). A multiplicação escalar por \mathbf{E}_f é transferida, no H.264/AVC, para a etapa de quantização que segue a etapa de transformadas. A etapa de transformadas pode ser implementada utilizando somente somas e deslocamentos.

A transformada Hadamard 4x4, definida em (7), é então aplicada ao bloco 4x4 de resíduos DC resultantes da aplicação da FDCT 2-D (“-1” na Figura 2.7), somente quando o tipo de predição é Intra 16x16. Em (7), \mathbf{W}_D representa o bloco de resíduos DC e \mathbf{Y}_D é o bloco de resíduos DC transformados pela Hadamard 4x4.

$$\mathbf{Y}_D = \begin{pmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{W}_D \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \end{pmatrix} / 2 \quad (7)$$

A transformada Hadamard 2x2, definida em (8), é aplicada aos blocos 2x2 de resíduos DC resultantes da aplicação da FDCT 2-D para os blocos de crominância 16 e 17 na Figura 2.7. Em (8), \mathbf{W}_D representa o bloco de resíduos DC e \mathbf{W}_{QD} é o bloco de resíduos DC transformados pela Hadamard 2x2.

$$\mathbf{W}_{QD} = \begin{pmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{W}_D \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \end{pmatrix} \quad (8)$$

2.4.4 Quantização Direta (Q)

A quantização direta (bloco Q da Figura 2.2) é aplicada somente no codificador H.264/AVC, realizando uma quantização escalar ao resultado dos coeficientes processados pela transformada direta. Este é o único módulo que gera perdas no processo de codificação do vídeo segundo o H.264/AVC. O fator de quantização controla a amplitude desta perda (variando a relação taxa-distorção do vídeo na saída) e depende do parâmetro de quantização QP (*Quantization Parameter*). O QP no H.264/AVC varia de 0 a 51, onde valores mais altos reduzem o número de bits utilizados para representar o vídeo, ao custo de uma perda da qualidade da imagem.

Os cálculos do bloco de quantização direta dependem do tipo de predição e se o elemento é de luminância e crominância. Em geral, as operações realizadas são uma multiplicação da entrada por uma constante, a soma do resultado por uma outra constante e um deslocamento da soma controlado por uma terceira constante. As constantes são controladas pelo QP, onde cada valor de QP está relacionado com um passo de quantização (Qstep). Os seis primeiros valores de Qstep são definidos pelo padrão como está apresentado na Tabela 2.2, sendo que os demais valores são derivados dos seis primeiros, ou seja, o $Qstep_{(6)}$ é igual $Qstep_{(0)} \times 2$, assim por diante.

Tabela 2.2: Relação entre QP e Qstep

QP	0	1	2	3	4	5	6	...	12
Qstep	0,625	0,6875	0,8125	0,875	1	1,125	1,25	...	2,5

Fonte: (AGOSTINI, 2007)

Para os elementos que foram processados apenas pela FDCT 2-D no bloco T, a quantização é definida em (9).

$$\begin{aligned} |Z_{(i,j)}| &= (|W_{(i,j)}| \cdot MF + f) \gg qbits \\ sign(Z_{(i,j)}) &= sign(W_{(i,j)}) \end{aligned} \quad (9)$$

Em (9), W_{ij} é o coeficiente resultante da FDCT 2-D, MF , definida em (10), é uma constante gerada a partir do fator de escala e do QP , f é uma constante definida pelo padrão em função da predição ter sido gerada pelo modo inter ou intra e do parâmetro de quantização utilizado. Por fim, $qbits$ indica o deslocamento que deve ocorrer na saída antes do cálculo ser finalizado.

$$MF = \frac{PF}{Qstep} \ll qbits \quad (10)$$

Em (10), PF é o fator de escala, $Qstep$ é o passo de quantização e $qbits$ é o mesmo deslocamento apresentado em (9). PF possui valores como a^2 , $ab/2$ ou $b^2/4$ dependendo da posição da amostra no bloco, sendo que a e b são as mesmas constantes definidas para o bloco T em (6). O valor de $qbits$ é função de QP e está apresentado em (11). A constante f mostrada em (9) é definida em (12).

$$qbits = 15 + \lfloor QP/6 \rfloor \quad (11)$$

$$\begin{aligned} f &= 2^{qbits} / 3 && \text{se a predição for intra} \\ f &= 2^{qbits} / 6 && \text{se a predição for inter} \end{aligned} \quad (12)$$

Para os coeficientes DC de crominância ou para os coeficientes de luminância codificados no modo Intra 16x16 é aplicada uma quantização distinta, definida em (13).

$$\begin{aligned} |Z_{D(i,j)}| &= (|Y_{D(i,j)}| \cdot MF_{(0,0)} + 2f) \gg (qbits + 1) \\ sign(Z_{D(i,j)}) &= sign(Y_{D(i,j)}) \end{aligned} \quad (13)$$

Esta quantização é bem parecida com a apresentada em (9), sendo as constantes MF , $qbits$ e f idênticas às definidas em (10), (11) e (12).

2.4.5 Quantização Inversa (QI)

A etapa de quantização inversa, ou *rescaling* (RICHARDSON, 2003), é aplicada no codificador e no decodificador de vídeo segundo o H.264/AVC. No decodificador, ela é aplicada após a etapa de decodificação de entropia. No codificador, ela é aplicada após a etapa de quantização direta. A quantização inversa realiza a correção de escala para o cálculo das transformadas sendo aplicada após as transformadas Hadamard 4x4 e 2x2 inversas, no caso de predição Intra 16x16 ou Croma, e sobre blocos de coeficientes quantizados, nos demais casos. Os mesmos parâmetros de quantização (QP e $Qstep$) são usados para esta etapa.

Para os elementos que foram processados apenas pela FDCT 2-D no bloco T, a quantização inversa é definida por (14).

$$W'_{(i,j)} = Z_{(i,j)} \cdot V_{(i,j)} \cdot 2^{\lfloor QP/6 \rfloor} \quad (14)$$

Em (14), $Z_{(i,j)}$ é o coeficiente quantizado, $V_{(i,j)}$ é uma constante gerada a partir do fator de escala ($PF_{(i,j)}$) e do parâmetro de quantização (QP), como em (15), onde $Qstep$ é o passo de quantização, o mesmo mostrado na Tabela 2.2.

$$V_{(i,j)} = (Qstep \cdot PF_{(i,j)} \cdot 64) \quad (15)$$

O fator de escala $PF_{(i,j)}$ é diferente em relação à quantização direta, mas também depende da posição da amostra no bloco. Na quantização inversa $PF_{(i,j)}$ pode assumir os valores a^2 , ab ou b^2 sendo que a e b são as mesmas constantes definidas para o bloco T em (6).

A quantização inversa para elementos DC de luminância que foram codificados usando predição Intra 16x16 está definida em (16), sendo que a constante $V_{(i,j)}$ foi definida em (15). A quantização inversa para elementos DC de crominância é definida em (17).

$$\begin{aligned} W'_{D(i,j)} &= W_{QD(i,j)} \cdot V_{(0,0)} \cdot 2^{\text{floor}(QP/6)} & (QP \geq 12) \\ W'_{D(i,j)} &= \left[W_{QD(i,j)} \cdot V_{(0,0)} \cdot 2^{1-\text{floor}(QP/6)} \right] \gg \lfloor QP/6 \rfloor & (QP < 12) \end{aligned} \quad (16)$$

$$\begin{aligned} W'_{D(i,j)} &= W_{QD(i,j)} \cdot V_{(0,0)} \cdot 2^{\lfloor QP/6 \rfloor - 1} & (QP \geq 6) \\ W'_{D(i,j)} &= \left[W_{QD(i,j)} \cdot V_{(0,0)} \right] \gg 1 & (QP < 6) \end{aligned} \quad (17)$$

2.4.6 Transformadas Inversas (TI)

O bloco TI das Figuras 2.2 e 2.3 é responsável pelas transformadas inversas dos coeficientes resultantes das etapas de transformadas (T) e quantização (Q) diretas. As operações das transformadas inversas são muito semelhantes às operações das transformadas diretas. As transformadas Hadamard 2x2 e Hadamard 4x4 são calculadas diretamente sobre os coeficientes DC provenientes da etapa de quantização, antes 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).

A transformada IDCT 2-D está definida em (18), onde \mathbf{X} é a matriz 4x4 de entrada, \mathbf{C}_i é a matriz IDCT inteira em uma dimensão, \mathbf{C}_i^T é a transposta da matriz IDCT e \mathbf{E}_i é a matriz de fatores de escala. O símbolo \otimes na equação indica uma multiplicação escalar. As letras a e b na matriz \mathbf{E}_i são as mesmas constantes definidas para a FDCT 2-D.

Do mesmo modo que para a FDCT 2-D, a multiplicação escalar por \mathbf{E}_i é transferida para o bloco de quantização inversa (QI), que é o passo que antecede a IDCT 2-D.

$$Y = C_i^T (X \otimes E_i) C_i = \begin{bmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{bmatrix} \left(\left[\begin{array}{c} X \\ \otimes \\ \left[\begin{array}{cccc} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{array} \right] \right] \right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} \end{bmatrix} \quad (18)$$

Para coeficientes de crominância ou de luminância quando a predição escolhida é Intra 16x16, a transformada Hadamard 4x4 inversa é aplicada sobre os coeficientes DC dos blocos de luminância, enquanto que, para os blocos de crominância, é aplicada a Hadamard 2x2 inversa sobre os blocos 2x2 de coeficientes DC de crominância.

O cálculo da Hadamard 4x4 inversa está apresentado em (19).

$$W_{QD} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} Z_D \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \quad (19)$$

Como pode ser observado em (19), o cálculo da Hadamard 4x4 inversa é muito parecido com o cálculo da Hadamard 4x4 direta que foi apresentada em (7). A única diferença entre elas está na divisão por dois, que existe na Hadamard direta, mas não existe na Hadamard 4x4 inversa.

A transformada Hadamard 2x2 inversa é aplicada apenas para amostras DC de crominância, sendo aplicada tanto para Cb quanto para Cr. O cálculo da Hadamard 2x2 inversa é idêntico ao cálculo da Hadamard 2x2 direta, que foi apresentado em (8). A ordem de processamento do bloco TI como um todo é exatamente a mesma apresentada na descrição do bloco T (Figura 2.7).

2.4.7 Filtro Redutor de Efeitos de Bloco

Este filtro é mandatório nos decodificadores, mas opcional nos codificadores de vídeo segundo o H.264/AVC. Este filtro está no laço de codificação inter-quadros, tendo como objetivo tornar mais suave o efeito de blocos do quadro reconstruído antes de ele ser usado para a predição de um macrobloco inter ou para ser exibido.

A novidade do filtro do H.264/AVC em relação aos padrões anteriores reside em sua adaptabilidade, ou seja, sua capacidade de distinguir uma aresta real da imagem, que não deve ser filtrada, de um efeito de bloco gerado por um elevado passo de quantização, que deve ser filtrado (RICHARDSON, 2003).

A operação de filtragem é feita sobre as bordas verticais e horizontais dos blocos 4x4 de um macrobloco já reconstruído (após o somatório dos resultados da transformada inversa e da predição utilizada no macrobloco). A ordem dos passos realizados pelo filtro é a seguinte (considere a Figura 2.8 como referência):

- 1) Filtrar as quatro bordas verticais do componente de luminância (**a**, **b**, **c** e **d**);
- 2) Filtrar as quatro bordas horizontais do componente de luminância (**e**, **f**, **g** e **h**);
- 3) Filtrar as duas bordas verticais de cada componente de crominância (**i** e **j**);
- 4) Filtrar as duas bordas horizontais de cada componente de crominância (**k** e **l**).



Figura 2.8: Ordem de filtragem das bordas de um macrobloco (ROSA, 2009).

A operação de filtragem modifica até três amostras de cada lado da borda de um bloco 4x4. A Figura 2.9 mostra as oito amostras de dois blocos 4x4 adjacentes (**p** e **q**) para o processo de filtragem de uma borda.

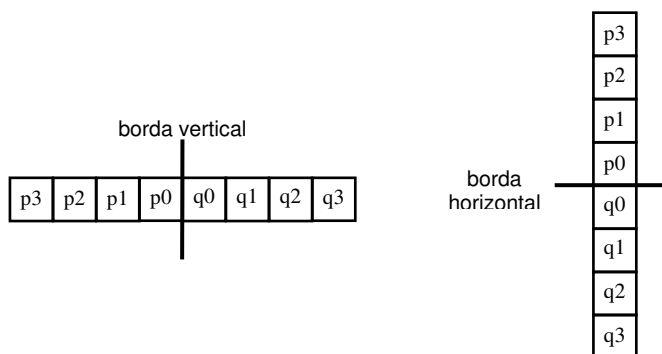


Figura 2.9: Amostras adjacentes para bordas verticais e horizontais (ROSA, 2009).

Como o filtro é adaptativo, a cada filtragem é aplicada uma “força” que depende do parâmetro de quantização, do modo de codificação dos blocos vizinhos e do gradiente das amostras da imagem através da borda. O filtro é responsável por uma melhora na qualidade subjetiva do vídeo reconstruído, especialmente para vídeos com baixa taxa de bits.

2.4.8 Codificação de Entropia

O processo de codificação de entropia descrito no padrão H.264/AVC utiliza-se basicamente de dois métodos: a codificação por códigos de comprimento variável (VLC, do inglês *Variable Length Coding*) e a codificação binária aritmética (BAC, do inglês *Binary Arithmetic Coding*). Cada bloco do codificador gera uma série de informações para posterior codificação, como modos de predição, tipo do macrobloco e os valores dos resíduos quantizados, por exemplo. Para cada uma destas informações é gerado um elemento sintático. O bloco de codificação de entropia trabalha sobre os elementos sintáticos para extrair estatísticas que correlacionam estes dados para representá-los com menor número de bits.

O H.264/AVC possui dois modos de codificação de entropia. O primeiro modo combina códigos de comprimento fixo, códigos *Exp-Golomb* e codificação de comprimento variável adaptativa ao contexto (CAVLC) aplicada aos blocos de resíduos quantizados (RICHARDSON, 2003). Este modo é mais simples e está presente em todos os perfis do padrão. O segundo modo utiliza codificação aritmética binária adaptativa ao contexto (CABAC) para codificar os resíduos quantizados (ao invés do CAVLC) e todas as informações relativas à codificação de macroblocos (modos de predição, vetores de movimento, etc). Este modo é mais complexo e pode ser usado somente nos perfis *main* e *high* do padrão.

Os códigos *Exp-Golomb* (*Exponential Golomb*) são códigos de comprimento variável que atribuem sequências de bits menores a elementos mais frequentemente utilizados, gerando sequências de bits maiores para elementos que aparecem com menos frequência. A Tabela 2.3 mostra um exemplo dos dez primeiros códigos gerados (Bit string) por um modo *Exp-Golomb*, a partir de um elemento de entrada denominado codeNum.

Tabela 2.3: Dez primeiros códigos *Exp-Golomb*.

codeNum	Bit string
0	1
1	010
2	011
3	00100
4	00101
5	00110
6	00111
7	0001000
8	0001001
9	0001010
...	...

Fonte: (ITU, 2007), p. 206.

A codificação de comprimento variável adaptativa ao contexto (CAVLC), presente em todos os perfis do padrão H.264/AVC, é aplicada somente aos resíduos resultantes da quantização direta (bloco Q) e foi desenvolvida especificamente para explorar as características dos blocos quantizados (RICHARDSON, 2003). A CAVLC atual como um VLC, assim como o Exp-Golomb, mas, leva em conta o contexto da codificação, ou seja, a fase em que o algoritmo de codificação se encontra e dos valores que já foram codificados.

A codificação aritmética binária adaptativa ao contexto (CABAC), presente somente nos perfis *main* e *high* é uma opção mais eficiente (e mais complexa) de codificação de entropia em comparação ao VLC (exp-Golomb e CAVLC) (RICHARDSON, 2003). Ela é baseada na codificação aritmética associada à seleção de modelos de probabilidade de ocorrência para cada elemento sintático de acordo com o contexto de codificação de cada elemento.

Neste capítulo, foram apresentados alguns conceitos de compressão de vídeo e uma introdução sobre o padrão H.264/AVC. O capítulo 3 apresenta com mais detalhe os conceitos relacionados à predição intra-quadro, presente nos codificadores compatíveis com o padrão H.264/AVC, por ser este o objeto desta dissertação.

3 A PREDIÇÃO INTRA-QUADRO

A predição intra-quadro é uma das inovações do padrão H.264/AVC. Esta predição é aplicada no domínio espacial, para todas as amostras de um macrobloco, tendo como objetivo diminuir a redundância espacial contida em um quadro do vídeo. O resultado da predição de um macrobloco é subtraído do macrobloco original para geração do resíduo a ser codificado. Este capítulo revisará os conceitos básicos sobre a predição intra-quadro no padrão H.264/AVC, foco principal desta dissertação, para contextualizar as decisões tomadas e fundamentar a arquitetura de *hardware* proposta.

3.1 Técnicas de Exploração da Redundância Espacial

Antes da publicação do padrão H.264/AVC, os padrões de compressão de imagens estáticas e vídeos usavam algumas técnicas simples para correlacionar dados em um mesmo quadro do vídeo, explorando a redundância espacial para compressão.

O padrão de compressão de imagens JPEG (*Joint Photographic Experts Group*) (ITU, 1992) prevê a predição de valores DC dos blocos 8x8 resultantes da transformada DCT e quantização, aplicando um cálculo baseado no DPCM (*Differential Pulse Code Modulation*) para explorar a correlação entre os blocos (GHANBARI, 2003). O padrão de compressão de vídeo MPEG-1 (ISO, 1993) usa este mesmo tipo de predição do JPEG para os coeficientes DC. O processo de predição é muito simples, sendo definido em (23).

$$DC_{pred} = DC_{cur} - DC_{prev} \quad (23)$$

Em (23), DC_{cur} é o coeficiente DC do bloco 8x8 já transformado e quantizado que está sendo codificado, DC_{prev} é o coeficiente DC do bloco 8x8 codificado anteriormente e DC_{pred} é a diferença entre os dois valores, que é enviada à codificação de entropia para compor o vídeo codificado (GHANBARI, 2003).

O anexo I do padrão H.263 (ITU, 2000), que compõe um conjunto de anexos conhecidos como H.263+ (GHANBARI, 2003), adota um modo de predição intra opcional chamado *Advanced Intra Coding Mode*. Esta predição possui conceitos similares à predição intra-quadro que foi adotada no H.264/AVC. Para a predição de um bloco 8x8 (menor tamanho de bloco usado no H.263), são usados os blocos vizinhos acima e à esquerda do bloco a ser codificado. Os coeficientes DC são preditos como a média dos coeficientes dos blocos acima e à esquerda. A predição dos coeficientes AC do bloco são geradas a partir de operações sobre as amostras dos blocos vizinhos, acima e à esquerda (ITU, 2000). Quando os blocos vizinhos não estiverem disponíveis, estas predições não podem ser realizadas (GHANBARI, 2003).

O padrão H.264/AVC propõe um processo de predição intra-quadro mais refinado: todas as amostras de um macrobloco são preditas com base nas amostras espaciais já reconstruídas de macroblocos vizinhos, como será detalhado nas próximas seções.

3.2 A Predição Intra-Quadro no H.264/AVC

No H.264/AVC, a predição intra-quadro é aplicada a cada macrobloco do tipo Intra, tanto para as amostras de luminância (Y) quanto para as amostras de croma (Cb e Cr). A predição de um macrobloco é gerada a partir de amostras já reconstruídas (porém não filtradas) localizadas nas fronteiras deste macrobloco e contidas dentro de um mesmo *slice* do quadro. Entende-se por amostras reconstruídas as amostras que já foram processadas pelo caminho de reconstrução T/Q/TI/QI (Transformadas e Quantização Diretas e Inversas), mas que não foram processadas pelo filtro de deblocação. Como mostrado no capítulo 2, a reconstrução é realizada no codificador para manter a coerência com a predição feita no decodificador (RICHARDSON, 2003).

Dois tipos de macrobloco I (Intra) são definidos no H.264/AVC: Intra 4x4 e Intra 16x16. No primeiro caso, as 16x16 amostras de luminância são divididas em blocos de 4x4 amostras para predição, como mostra a Figura 3.1. Os blocos 4x4 são numerados de acordo com sua ordem de processamento, que é a mesma ordem de processamento dos blocos de transformadas e quantização. No segundo caso, o macrobloco é predito por inteiro, sem subdivisão. Somente um tipo de predição é definido para ambos os blocos 8x8 de croma (Cb e Cr), quando a sub-amostragem 4:2:0 é usada, não havendo subdivisão dos blocos para predição.

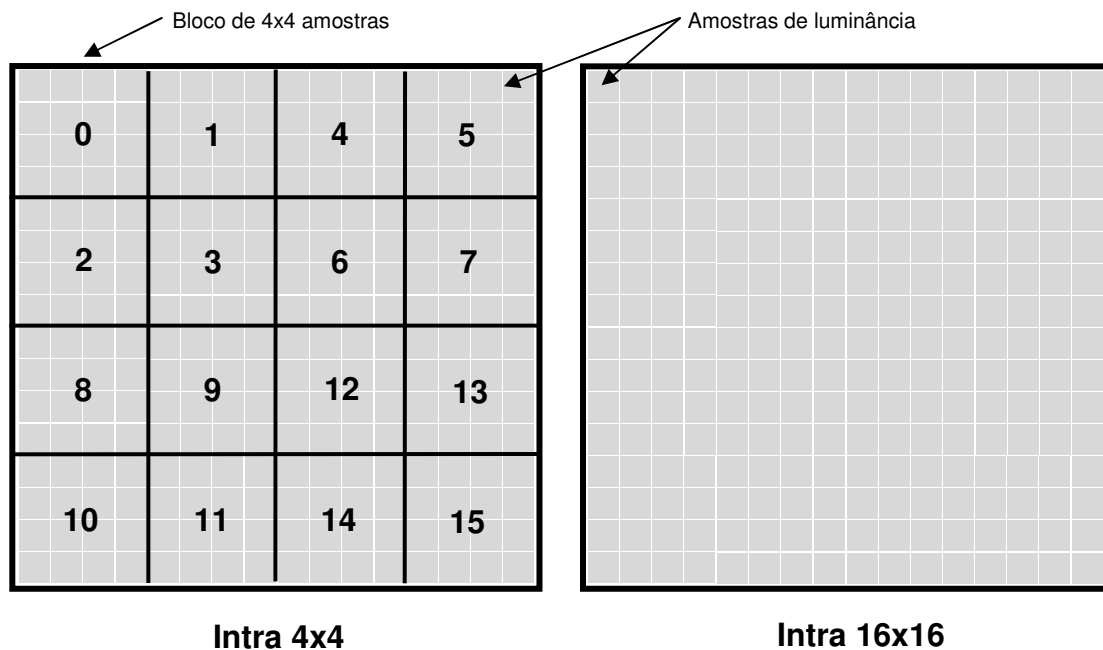


Figura 3.1: Tipos de macrobloco Intra-Quadro.

Cada tipo de predição (Intra 4x4 ou Intra 16x16) contém diferentes modos para gerar o macrobloco predito. Estes modos procuram reproduzir diferentes padrões espaciais contidos em um quadro do vídeo. Um macrobloco predito muito similar ao macrobloco

original pode reduzir significativamente o valor dos resíduos a serem codificados e, por consequência, o número de bits para codificar o macrobloco em questão. O diferente particionamento (Intra 4x4 ou Intra 16x16) determina a granularidade com que é feita a predição. O modo Intra 4x4 possui granularidade mais fina que o tipo Intra 16x16, porém gera mais bits de informação dos modos de predição a serem enviados para o decodificador, como será detalhado nas próximas seções.

A figura 3.2 mostra um detalhe da aplicação da predição intra-quadro em um quadro da sequência de vídeo *Foreman*, na resolução QCIF (176x144). Foi usada a predição Intra 4x4 em todo quadro, sendo que as linhas sobrepostas na Figura 3.2 representam as direções dos diferentes modos de predição, detalhados na seção 3.2.1. A Figura 3.3 mostra a diferença visual entre o quadro original e o quadro de resíduo resultante da aplicação da predição Intra 4x4. Na imagem do resíduo a cor cinza claro representa uma diferença de zero entre a amostra original e a amostra predita. Tons de cinza mais claros ou mais escuros representam diferenças maiores. Somente as informações de resíduo (após transformados e quantizados) e modos de predição são enviados no *bitstream*.

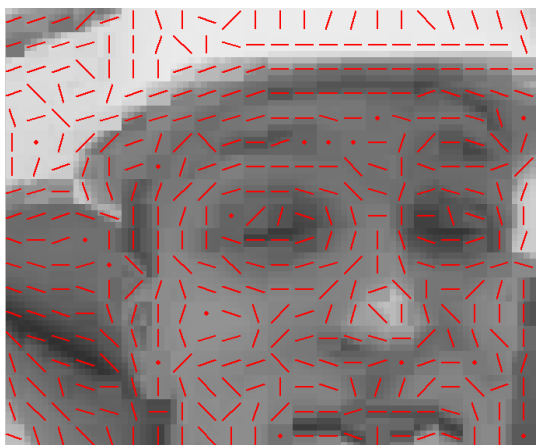


Figura 3.2: Detalhe da aplicação da predição Intra 4x4 em um quadro da sequência de vídeo *Foreman* (QCIF).

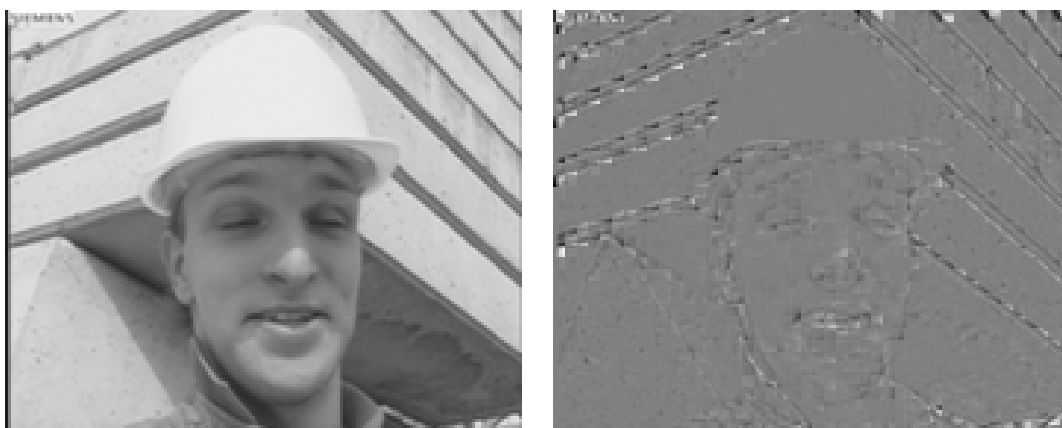


Figura 3.3: Resultado da predição Intra 4x4. Quadro original (esquerda); Resíduo da predição Intra 4x4 (direita).

3.2.1 Modos de Predição Intra 4x4

O tipo Intra 4x4 define 9 modos de predição que abrangem padrões horizontais, verticais e diagonais com diferentes ângulos de inclinação. Esta predição é muito adequada para regiões da imagem com maior nível de detalhe. A Figura 3.4 mostra as amostras vizinhas utilizadas para predição (A-M) e o bloco 4x4 resultante da etapa de predição, em cinza, juntamente com a numeração utilizada para identificar cada amostra do bloco 4x4. A Figura 3.5 ilustra as direções dos 9 modos, usando a mesma convenção da Figura 3.4.

M	A	B	C	D	E	F	G	H
I	0,0	0,1	0,2	0,3				
J	1,0	1,1	1,2	1,3				
K	2,0	2,1	2,2	2,3				
L	3,0	3,1	3,2	3,3				

Figura 3.4: Identificação das amostras para a predição Intra 4x4.

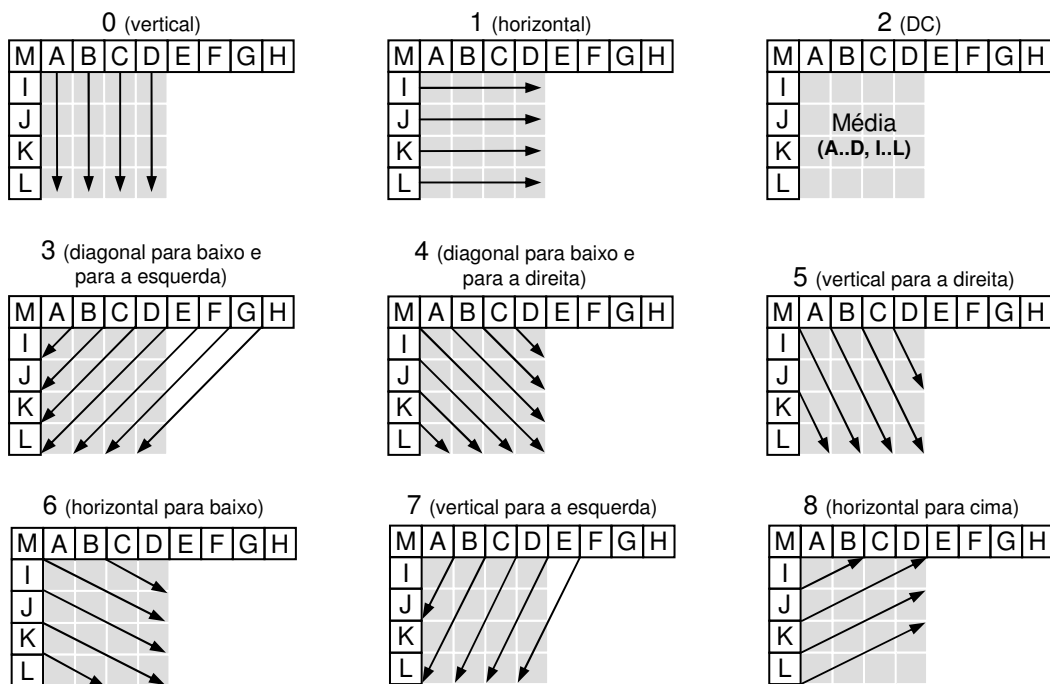


Figura 3.5: Modos de predição Intra 4x4 (AGOSTINI, 2007).

Nos modos 0 (vertical) e 1 (horizontal) o bloco predito é construído pela cópia das amostras vizinhas A-D e I-L (Figura 3.5), respectivamente. O modo 2 (DC) faz uma média das amostras A-D e I-L e copia o resultado para todas amostras do bloco predito. Os modos diagonais (3 a 8 na Figura 3.5) são calculados por interpolações lineares usando dos valores das amostras vizinhas, dependendo das direções das setas e da posição da amostra no bloco predito, como mostrado na Tabela 3.1. Esta tabela, baseada em (HUANG, 2005), detalha todas as operações feitas em cada modo de predição Intra 4x4.

Tabela 3.1: Modos de Predição Intra 4x4 (I4MB), baseada em (HUANG, 2005).

Modo I4MB	L	K	J	I	M	A	B	C	D	E	F	G	H	Soma	Desloca	Posições (y, x)
0						1								0	>> 0	(0,0) (1,0) (2,0) (3,0)
							1							0	>> 0	(0,1) (1,1) (2,1) (3,1)
								1						0	>> 0	(0,2) (1,2) (2,2) (3,2)
									1					0	>> 0	(0,3) (1,3) (2,3) (3,3)
1				1										0	>> 0	(0,0) (0,1) (0,2) (0,3)
			1											0	>> 0	(1,0) (1,1) (1,2) (1,3)
		1												0	>> 0	(2,0) (2,1) (2,2) (2,3)
	1													0	>> 0	(3,0) (3,1) (3,2) (3,3)
2	1	1	1	1		1	1	1	1				4	>> 3	Todas posições	
3						1	2	1						2	>> 2	(0,0)
							1	2	1					2	>> 2	(0,1) (1,0)
								1	2	1				2	>> 2	(0,2) (1,1) (2,0)
									1	2	1			2	>> 2	(0,3) (1,2) (2,1) (3,0)
										1	2	1		2	>> 2	(1,3) (2,2) (3,1)
											1	2	1	2	>> 2	(2,3) (3,2)
4												1	3	2	>> 2	(3,3)
	1	2	1											2	>> 2	(3,0)
		1	2	1										2	>> 2	(2,0) (3,1)
			1	2	1									2	>> 2	(1,0) (2,1) (3,2)
				1	2	1								2	>> 2	(0,0) (1,1) (2,2) (3,3)
					1	2	1							2	>> 2	(0,1) (1,2) (2,3)
5						1	2	1						2	>> 2	(1,3)
							1	2	1					2	>> 2	(0,3)
					1	1								1	>> 1	(0,0) (2,1)
						1	1							1	>> 1	(0,1) (2,2)
							1	1						1	>> 1	(0,2) (2,3)
								1	1					1	>> 1	(0,3)
		1	2	1										2	>> 2	(3,0)
			1	2	1									2	>> 2	(2,0)
6				1	2	1								2	>> 2	(1,0) (3,1)
					1	2	1							2	>> 2	(1,1) (3,2)
						1	2	1						2	>> 2	(1,2) (3,3)
							1	2	1					2	>> 2	(1,3)
								1	2	1				2	>> 2	(0,3)
									1	2	1			2	>> 2	(3,0)
										1	2	1		2	>> 2	(2,0) (3,2)
											1	2	1	2	>> 2	(1,0) (2,2)
7														1	>> 1	(0,0) (1,2)
	1	2	1											2	>> 2	(3,1)
		1	2	1										2	>> 2	(2,1) (3,3)
			1	2	1									2	>> 2	(1,1) (2,3)
				1	2	1								2	>> 2	(0,1) (1,3)
					1	2	1							2	>> 2	(0,2)
						1	2	1						2	>> 2	(0,3)
							1	1						1	>> 1	(0,0)
8														1	>> 1	(0,1) (2,0)
														1	>> 1	(0,2) (2,1)
														1	>> 1	(0,3) (2,2)
														1	>> 1	(2,3)
						1	2	1						2	>> 2	(1,0)
							1	2	1					2	>> 2	(1,1) (3,0)
								1	2	1				2	>> 2	(1,2) (3,1)
									1	2	1			2	>> 2	(1,3) (3,2)
									1	2	1		2	>> 2	(3,3)	
8	1													0	>> 0	Demais posições
	1	1												1	>> 1	(1,2) (2,0)
		1	1											1	>> 1	(0,2) (1,0)
			1	1										1	>> 1	(0,0)
	3	1												2	>> 2	(1,3) (2,1)
	1	2	1											2	>> 2	(0,3) (1,1)
		1	2	1									2	>> 2	(0,1)	

Na Tabela 3.1, as letras L-H representam as amostras vizinhas reconstruídas e os números associados a cada letra são os fatores multiplicativos. Ou seja, cada linha da Tabela 3.1 representa uma operação do modo Intra 4x4 (ou I4MB) que gera a amostra predita de posição (y,x) no bloco 4x4. Note que ao final de cada operação um arredondamento é aplicado. Este consiste na soma do valor resultante a uma constante e um deslocamento para direita no resultado desta soma.

Observa-se na Tabela 3.1 que o resultado de algumas operações gera o valor de mais de uma amostra predita. Por exemplo, para as amostras preditas no modo 3, com as posições (0,1) e (1,0) no bloco 4x4, o valor é determinado por $(B + 2C + D + 2) \gg 2$. Além disso, esta mesma operação é aplicada para gerar a amostra (0,3) do modo 4, a amostra (1,3) do modo 5, e as amostras (1,1) e (3,0) do modo 7. Ou seja, percorrendo a tabela no sentido vertical e observando as operações comuns entre os diferentes modos percebe-se que existe muito compartilhamento de operações no cálculo das amostras preditas para o tipo Intra 4x4. Este fator influenciou as decisões arquiteturais do preditor intra-quadro, descritas no capítulo 4 desta dissertação.

Nesta seção, assumimos o caso geral que todas as amostras vizinhas estão disponíveis para o cálculo da predição. Veremos na seção 3.2.4 que existem situações que este caso não se aplica e como eles são tratados no padrão H.264/AVC.

3.2.2 Modos de Predição Intra 16x16

A predição Intra 16x16 gera um macrobloco predito de tamanho 16x16 amostras de luminância, a partir das 32 amostras de luminância vizinhas contidas na borda dos macroblocos acima e à esquerda do macrobloco a ser codificado (H e V na Figura 3.6). São definidos 4 modos de predição, como ilustrado na Figura 3.6.

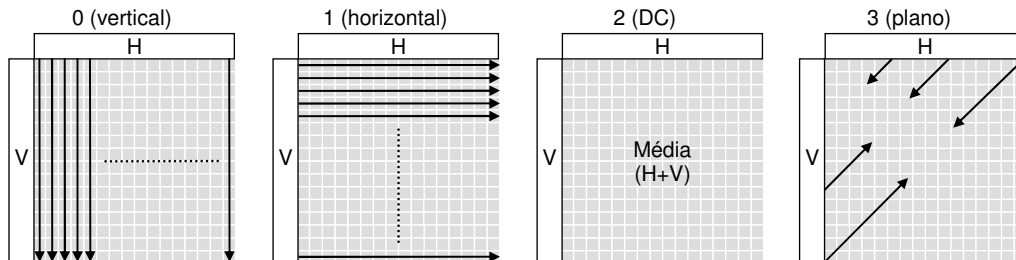


Figura 3.6: Modos de predição Intra 16x16 (AGOSTINI, 2007).

No modo 0 (vertical) o macrobloco predito é construído pela cópia das amostras vizinhas contidas em H. O modo 1 (horizontal) é similar ao modo 0, mas copia as amostras em V (Figura 3.6). O modo 2 (DC) faz uma média das amostras contidas em H e V, como em (24), e copia o resultado para todas amostras do macrobloco predito.

$$DC = \left(\sum_{x'=0}^{15} p[x', -1] + \sum_{y'=0}^{15} p[-1, y'] + 16 \right) \gg 5 \quad (24)$$

Em (24), $p[y,x]$ é uma amostra localizada na linha y e coluna x do macrobloco, onde $[0,0]$ representa a amostra do seu canto superior esquerdo. Quando o índice é “-1” significa que a amostra está no macrobloco vizinho (esquerdo ou superior), que já foi

codificado e reconstruído. **DC** é o valor resultante que é copiado para todas as amostras do macrobloco predito.

O modo 3 (plano) é o mais complexo de ser calculado, pois necessita a geração prévia de três parâmetros (**a**, **b** e **c**) antes do cálculo da predição para cada amostra do macrobloco predito, definida em (25).

$$P[y, x] = \text{Clip}((a + b \cdot (x - 7) + c \cdot (y - 7) + 16) \gg 5) \quad (25)$$

Em (25), **P[y,x]** é a amostra predita no modo plano, **Clip** é uma função de saturação para adequar os valores na faixa de representação da imagem (0 a 255, para amostras de 8 bits), e os parâmetros **a**, **b** e **c** são definidos em (26), (27) e (28), respectivamente.

$$a = 16 \cdot (p[-1,15] + p[15,-1]) \quad (26)$$

$$b = (5 \cdot H + 32) \gg 6 \quad (27)$$

$$c = (5 \cdot V + 32) \gg 6 \quad (28)$$

Em (26), **p[y,x]** possui o mesmo significado de **p[y,x]** mostrado em (24). Os valores de **H** e **V** em (27) e (28) estão definidos em (29) e (30), respectivamente. Em (29) e (30), **p[y,x]** possui o mesmo significado de **p[y,x]** em (24).

$$H = \sum_{x'=0}^7 (x'+1) \cdot (p[-1,8+x'] - p[-1,6-x']) \quad (29)$$

$$V = \sum_{y'=0}^7 (y'+1) \cdot (p[8+y',-1] - p[6-y',-1]) \quad (30)$$

3.2.3 Modos de Predição para Crominância

Os modos de predição para amostras de crominância são semelhantes aos modos Intra 16x16, porém o número dos modos é alterado, como mostrado na Figura 3.7. No caso de sub-amostragem 4:2:0, usado neste trabalho, a predição é aplicada a blocos 8x8 de crominância azul (Cb) e vermelha (Cr), a partir de 16 amostras vizinhas (H e V na Figura 3.7). O mesmo modo é aplicado para ambos canais de crominância Cb e Cr.

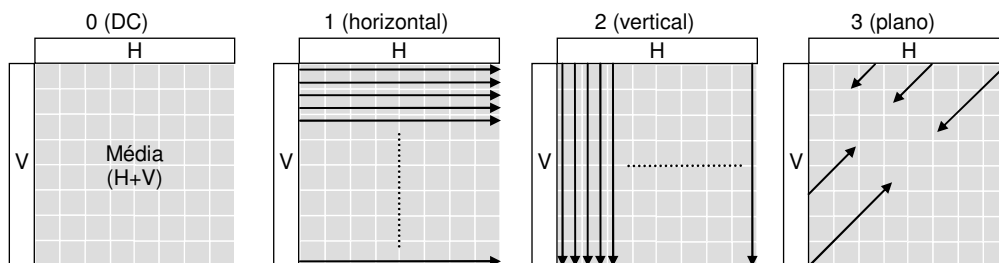


Figura 3.7: Modos de predição para blocos 8x8 de crominância.

O modo DC para crominância é similar a (24), mas calculado sobre as 16 amostras vizinhas (8 em H, 8 em V), portanto, os valores de arredondamento e deslocamento também são diferentes: soma a 8 e deslocamento de 4 bits.

O modo plano para crominância é similar ao modo plano do modo Intra 16x16. A predição de uma amostra é definida em (31), considerando a relação de sub-amostragem 4:2:0, sendo os parâmetros **a**, **b** e **c** definidos em (32), (33) e (34), respectivamente.

$$P[y,x] = \text{Clip}((a + b \cdot (x - 3) + c \cdot (y - 3) + 16) \gg 5) \quad (31)$$

$$a = 16 \cdot (p[-1,7] + p[7,-1]) \quad (32)$$

$$b = (34 \cdot H + 32) \gg 6 \quad (33)$$

$$c = (34 \cdot V + 32) \gg 6 \quad (34)$$

Em (32), **p[y,x]** é uma amostra localizada na linha **y** e coluna **x** do canal de crominância do macrobloco, onde [0,0] representa a amostra do canto superior esquerdo. Os valores de **H** e **V** em (33) e (34) estão definidos em (35) e (36), respectivamente. Em (35) e (36), **p[y,x]** possui o mesmo significado de **p[y,x]** em (31).

$$H = \sum_{x'=0}^3 (x'+1) \cdot (p[4+x',-1] - p[2-x',-1]) \quad (35)$$

$$V = \sum_{y'=0}^3 (y'+1) \cdot (p[-1,4+y'] - p[-1,2-y']) \quad (36)$$

3.2.4 Cálculo da Disponibilidade dos Blocos Vizinhos

Nem sempre as amostras vizinhas (já reconstruídas) estão disponíveis para predição intra-quadro. Isto ocorre, por exemplo, no caso dos macroblocos localizados na fronteira do quadro. Em geral, quando as amostras vizinhas não estão disponíveis para a predição de um determinado modo, este modo não é aplicado. As exceções são os modos DC Intra 4x4 e Intra 16x16, como veremos a seguir.

Antes disto, vamos revisar a convenção usada no H.264/AVC para indicar os macroblocos (ou blocos) vizinhos de um macrobloco (ou bloco). Ela é mostrada na Figura 3.8, onde **Curr** é o macrobloco (ou bloco) que está sendo predito no momento, **A**, **B**, **C** e **D** são seus macroblocos (ou blocos) vizinhos.

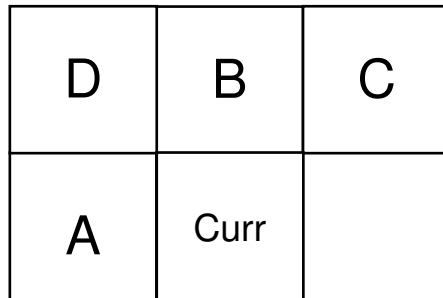


Figura 3.8: Convenção para indicar macroblocos ou blocos vizinhos.

Os macroblocos dentro de cada *slice* do quadro são numerados e organizados na ordem *raster*. Neste trabalho, usou-se apenas um *slice* por quadro, sendo que os macroblocos são numerados de 0 até o número máximo de macroblocos de um quadro e organizados na ordem *raster*. Nos seguintes casos os macroblocos (ou blocos) vizinhos não estão disponíveis para a predição intra-quadro:

- Para todos os macroblocos (ou blocos) **Curr** localizados nas fronteiras do quadro, todos vizinhos (A, B, C ou D) que excedem o quadro são marcados como não-disponíveis;
- Quando o macrobloco (ou bloco) vizinho está localizado em um *slice* diferente do *slice* que contém **Curr**, este macrobloco (ou bloco) vizinho é marcado como não-disponível;
- Quando o macrobloco (ou bloco) vizinho é do tipo Inter e **constrained_intra_pred_flag** é igual a 1, este também é marcado como não-disponível. Esta *flag* controla para que nenhum macrobloco do tipo Intra seja predito usando amostras reconstruídas de macroblocos do tipo Inter.

As operações anteriormente citadas nas seções 3.2.1, 3.2.2 e 3.2.3 consideram o caso em que todas as amostras vizinhas necessárias para predição de cada modo estão disponíveis. Na maioria dos casos, quando a vizinhança necessária para o processamento de um modo de predição não está disponível, esta predição não é realizada. As exceções a esta regra são:

- Na predição DC (Intra 4x4 e Intra 16x16), quando somente uma vizinhança é disponível (**A** ou **B**), a média é feita somente sobre as demais amostras do macrobloco (ou bloco) vizinho disponível, como em (37) e (38) para o Intra 4x4 e (40) e (41) para o Intra 16x16. Neste caso o valor do arredondamento é a metade do caso normal. Se nenhuma das vizinhanças é disponível, todas amostras recebem o valor 128 (centro da escala, 0-255), como em (39) e (42). Este é o único modo que não precisa de nenhuma vizinhança para ser gerado, sendo sempre usado no primeiro macrobloco;

$$DC_{4x4} = (I + J + K + L + 2) \gg 2 \quad (37)$$

$$DC_{4x4} = (A + B + C + D + 2) \gg 2 \quad (38)$$

$$DC_{4x4} = 128 \quad (39)$$

$$DC_{16x16} = (\sum_{x'=0}^{15} p[x', -1] + 8) \gg 4 \quad (40)$$

$$DC_{16x16} = (\sum_{y'=0}^{15} p[-1, y'] + 8) \gg 4 \quad (41)$$

$$DC_{16x16} = 128 \quad (42)$$

- Nas predições diagonais Intra 4x4 que usam a vizinhança C, para blocos 4x4 diferentes de 3 ou 11 (Figura 3.1), se a vizinhança C não está disponível, é feita uma extrapolação da amostra mais à direita da vizinhança B, ou seja, esta amostra vizinha é replicada para gerar as amostras da vizinhança C.

3.2.5 Sinalização dos Modos de Predição Intra-Quadro no Vídeo Codificado

O elemento sintático **mb_type** indica no bitstream (vídeo codificado) qual é o tipo de cada macrobloco. No caso de macroblocos do tipo Intra, **mb_type** assume valores de 0 a 25, como mostrado na Tabela 3.2.

Tabela 3.2: Tipos de macrobloco para *slices* I

mb_type	Nome do mb_type	transform_size_8x8_flag	MbPartPredMode (mb_type, 0)	Intra16x16PredMode	CodedBlockPatternChroma	CodedBlockPatternLuma
0	I_NxN	0	Intra_4x4	na	Eq. 7-34	Eq. 7-34
0	I_NxN	1	Intra_8x8	na	Eq. 7-34	Eq. 7-34
1	I_16x16_0_0_0	na	Intra_16x16	0	0	0
2	I_16x16_1_0_0	na	Intra_16x16	1	0	0
3	I_16x16_2_0_0	na	Intra_16x16	2	0	0
4	I_16x16_3_0_0	na	Intra_16x16	3	0	0
5	I_16x16_0_1_0	na	Intra_16x16	0	1	0
6	I_16x16_1_1_0	na	Intra_16x16	1	1	0
7	I_16x16_2_1_0	na	Intra_16x16	2	1	0
8	I_16x16_3_1_0	na	Intra_16x16	3	1	0
9	I_16x16_0_2_0	na	Intra_16x16	0	2	0
10	I_16x16_1_2_0	na	Intra_16x16	1	2	0
11	I_16x16_2_2_0	na	Intra_16x16	2	2	0
12	I_16x16_3_2_0	na	Intra_16x16	3	2	0
13	I_16x16_0_0_1	na	Intra_16x16	0	0	15
14	I_16x16_1_0_1	na	Intra_16x16	1	0	15
15	I_16x16_2_0_1	na	Intra_16x16	2	0	15
16	I_16x16_3_0_1	na	Intra_16x16	3	0	15
17	I_16x16_0_1_1	na	Intra_16x16	0	1	15
18	I_16x16_1_1_1	na	Intra_16x16	1	1	15
19	I_16x16_2_1_1	na	Intra_16x16	2	1	15
20	I_16x16_3_1_1	na	Intra_16x16	3	1	15
21	I_16x16_0_2_1	na	Intra_16x16	0	2	15
22	I_16x16_1_2_1	na	Intra_16x16	1	2	15
23	I_16x16_2_2_1	na	Intra_16x16	2	2	15
24	I_16x16_3_2_1	na	Intra_16x16	3	2	15
25	I_PCM	na	na	na	na	na

Fonte: ITU, 2007. p. 91.

Os valores 1 a 24 de **mb_type** correspondem ao tipo Intra 16x16. Neste caso, o modo de predição (Intra16x16PredMode) é codificado no próprio **mb_type**, combinado a valores do padrão de codificação do bloco (CodedBlockPatternLuma e CodedBlockPatternChroma). Os parâmetros CBP (*Coded Block Pattern*) indicam quando existe algum coeficiente válido em um bloco 8x8 resultante da etapa quantização, ou se para algum bloco 8x8 todos os coeficientes tem valor zero (ITU, 2007). O valor 25 indica o I_PCM, que é um tipo de macrobloco especial onde nenhuma codificação é aplicada, ou seja, as amostras são colocadas diretamente no bitstream. Este modo não é tratado neste trabalho. Uma vez gerado o valor bruto de **mb_type**, este elemento sintático é codificado usando códigos *Exp-Golomb*.

O valor 0 de **mb_type** indica que o macrobloco é do tipo Intra 4x4 ou Intra 8x8 (não usado neste trabalho por ser particular dos perfis *High*). Neste caso, os modos de predição para cada um dos 16 blocos 4x4 não são enviados diretamente no bitstream, mas sim codificados usando uma lógica de modo mais provável. O modo mais provável de um bloco 4x4 é o menor valor dentre os modos escolhidos para os blocos 4x4 vizinhos **A** e **B** (veja Figura 3.8). Caso algum destes blocos vizinhos não estiver disponível, o modo mais provável é o modo 2 (DC). O algoritmo para codificação dos modos 4x4 é executado para cada bloco 4x4 e está descrito a seguir, em pseudocódigo:

```

modo_mais_provável ← min( modo_vizinho_A , modo_vizinho_B )
Se ( modo_escolhido = modo_mais_provável )
    prev_intra4x4_pred_mode = 1;
    Insere a flag prev_intra4x4_pred_mode no bitstream
Senão
    prev_intra4x4_pred_mode = 0;
Se ( modo_escolhido > modo_mais_provável )
    rem_intra4x4_pred_mode = modo_escolhido - 1;
Senão
    rem_intra4x4_pred_mode = modo_escolhido;
Fim Se
Insere a flag rem_intra4x4_pred_mode no bitstream
Fim Se

```

O elemento sintático **rem_intra4x4_pred_mode** representa a diferença do modo escolhido para o modo mais provável de um bloco 4x4. Deste modo, somente 8 valores (3 bits) são usados para sinalizar os 9 modos de predição Intra 4x4 (RICHARDSON, 2003). A tabela 3.3 fornece um exemplo para o cálculo do elemento sintático **rem_intra4x4_pred_mode**.

O modo de predição para crominância é sinalizado no bitstream através do elemento sintático **intra_chroma_pred_mode**, sendo que seu valor (0 a 3) é codificado usando códigos *Exp-Golomb* (ITU, 2007).

Tabela 3.3: Escolha do modo Intra 4x4 (modo mais provável = 1).

rem_intra4x4_pred_mode	Modo de predição para um bloco 4x4
0	0
1	2
2	3
3	4
4	5
5	6
6	7
7	8

Fonte: RICHARDSON, 2003, p. 183.

3.3 Processo de Codificação Intra-Quadro

No processo de decodificação de vídeo segundo o padrão H.264/AVC, a predição intra-quadro é aplicada conforme o tipo do macrobloco (Intra 4x4 ou Intra 16x16) e os modos de predição sinalizados no *bitstream*, como foram detalhados na seção 3.2. O decodificador de vídeo somente deve gerar o macrobloco predito que for sinalizado no *bitstream* pelo codificador, sendo este processo de decodificação definido completamente pelo padrão H.264/AVC (ITU, 2007).

Fica a cargo do codificador de vídeo a tomada de decisão de qual é o melhor modo a ser codificado para cada macrobloco. Como foi mencionado no capítulo 2, este tipo de decisão não é normatizada, sendo que é livre a escolha do algoritmo de decisão dos modos de predição. A Figura 3.9 ilustra o processo de codificação intra-quadro.

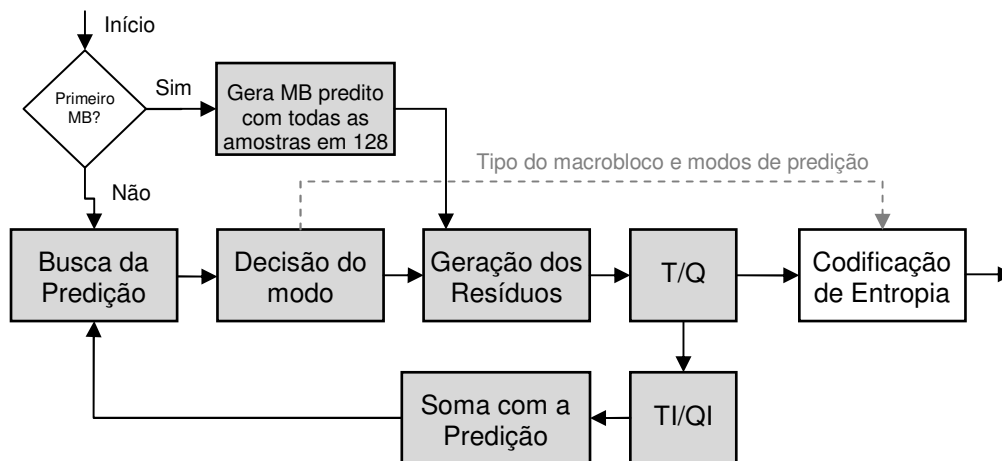


Figura 3.9: Processo de codificação intra-quadro.

O processo descrito na Figura 3.9 se baseia em uma etapa de busca da predição intra-quadro para cada macrobloco, seguido da decisão de qual modo de predição será usado para codificá-lo. No caso especial do primeiro macrobloco Intra a ser codificado em um *slice*, um tipo de predição especial é utilizado: um macrobloco (MB) predito é gerado com todas as amostras com o valor 128 (meio da escala de 8 bits, ou seja, de 0 a 255). Tendo o modo escolhido (ou no caso do primeiro macrobloco, a predição

especial), o resíduo é gerado pela subtração das amostras do macrobloco original com as amostras do macrobloco predito. Este macrobloco é dividido em blocos 4x4 para serem processadas pelo bloco T/Q (transformadas e quantização). Os coeficientes resultantes da etapa de quantização e as informações de tipo do macrobloco e modos de predição são posteriormente codificados pelo bloco de codificação de entropia. Os coeficientes quantizados são ainda processados pelo bloco TI/QI (Transformadas e Quantização Inversas) para serem somados às amostras preditas correspondentes no macrobloco predito, gerando o macrobloco reconstruído, que será usado para predição de outros macroblocos. Esta reconstrução é o mesmo processo feito no decodificador e garante que as predições feitas no codificador serão as mesmas realizadas na decodificação.

O processo descrito na Figura 3.9 é utilizado tanto para o tipo Intra 4x4 quanto para o tipo Intra 16x16. No Intra 16x16, um modo de predição é escolhido para cada macrobloco, gerando um macrobloco de resíduos. Os resíduos são divididos em blocos 4x4 para serem processados pelos blocos T/Q e TI/QI e soma com a predição. No caso do Intra 4x4, é escolhido um modo de predição localmente para cada bloco 4x4, sendo que este bloco 4x4 é processado pelos módulos de geração de resíduos, T/Q, TI/QI e soma com a predição. Deste modo, o tipo Intra 16x16 usa como referência amostras reconstruídas de macroblocos vizinhos. Já no tipo Intra 4x4 cada bloco 4x4 usa como referência as amostras reconstruídas dos blocos 4x4 anteriormente processados. Isto forma um laço de codificação que envolve os módulos destacados em cinza na Figura 3.9. A codificação de macroblocos intra-quadro do H.264/AVC insere latência no processo de codificação de vídeo, pois torna impraticável o *pipeline* no nível de macrobloco. Diferentemente dos padrões anteriores de compressão de vídeo, onde as etapas de busca da predição e de codificação do bloco (transformadas, quantização, VLC) eram claramente separadas, o H.264/AVC usa as amostras reconstruídas de um mesmo quadro para predição, não sendo possível o *pipeline* no nível de macrobloco (HUANG, 2005). O único módulo não envolvido no laço de codificação intra-quadro é a codificação de entropia, que pode trabalhar em *pipeline* com os demais módulos. Tais considerações relacionadas à arquitetura serão tratadas no capítulo 4.

Na Figura 3.9, a etapa de seleção da predição consiste em gerar os modos de predição intra-quadro definidos pelo padrão. Na literatura são encontradas duas classes de algoritmos para esta etapa: i) algoritmos de busca completa (*full-search*): calculam todos os modos de predição intra-quadro possíveis; ii) algoritmos rápidos: usam alguma heurística, baseada na informação do vídeo a ser codificado, para selecionar somente alguns modos de predição intra-quadro de forma a diminuir a complexidade da decisão do modo. Os trabalhos de Pan (PAN, 2004), Chang (CHANG, 2007), Li (LI, 2007) e Tsai (TSAI, 2008) propõem algoritmos rápidos para selecionar somente alguns modos de predição intra-quadro a serem comparados na etapa de decisão do modo. Os trabalhos de Suh (SUH, 2005) e Kuo (KUO, 2008) implementam o algoritmo de busca completa. A arquitetura proposta neste trabalho também implementa o algoritmo de busca completa, como será detalhado no capítulo 4.

A etapa de decisão do modo (Figura 3.9) avalia e escolhe um dos tipos e modos de predição gerados pela etapa de busca, usando uma determinada técnica de otimização taxa-distorção (como foi descrito no capítulo 2). No caso da codificação intra-quadro, as seguintes decisões devem ser tomadas para cada macrobloco a ser codificado:

- Escolha de um dos 9 modos de predição Intra 4x4 para cada um dos 16 blocos 4x4;

- Escolha de um dos 4 modos de predição Intra 16x16;
- Escolha de um dos 4 modos de predição para crominância;
- Decisão de qual o particionamento do macrobloco (sinalizado pelo **mb_type**): Intra 4x4 ou Intra 16x16.

Neste trabalho, as três primeiras escolhas são feitas usando SAD como critério de similaridade. Esta abordagem fornece uma aproximação da otimização taxa-distorção, pois seleciona o modo que resulta em resíduos próximos dos mínimos. A expectativa é de que esses valores reduzidos de resíduos resultem em menor taxa de informação codificada. Por outro lado, o SAD é mais simples para implementar em *hardware*, o que permite atingir o desempenho para o processamento de vídeos de alta definição em tempo-real.

Já a decisão de qual tipo de macrobloco utilizar (Intra 4x4 ou Intra 16x16) não só depende da magnitude dos resíduos como da informação de modo de predição gerada. Se somente o SAD for utilizado para esta escolha, o tipo Intra 4x4 sempre será escolhido, pois a menor granularidade e o maior número de modos de predição deste tipo favorecem a geração de resíduos de menor magnitude, diminuindo o valor do SAD. O que favorece a escolha do modo Intra 16x16 é o menor número de bits usado para codificação do modo de predição. Por exemplo, no tipo Intra 16x16 a codificação dos modos é enviada no **mb_type** em valores de 1 a 24. Depois da codificação Exp-Golomb esta informação gera, no pior caso, 9 bits/macrobloco. O tipo Intra 4x4 gera, no melhor caso, 16 bits/macrobloco (1 bit por bloco 4x4) no caso de ser escolhido sempre o modo mais provável. O pior caso pode chegar a 4 bits por bloco 4x4, totalizando 64 bits/macrobloco, sete vezes mais informação de modo que o Intra 16x16. Esta decisão depende ainda do cálculo do número de bits/macrobloco para codificar os resíduos, sendo que isto depende do método de codificação de entropia escolhido (CAVLC ou CABAC).

Devido à complexidade da decisão Intra 4x4/Intra 16x16 para codificação em tempo-real, por esta ser uma tarefa de controle do codificador e para delimitar o escopo desta dissertação de mestrado, este processo não será tratado neste trabalho. Outros trabalhos em andamento no grupo de pesquisa buscam encontrar uma heurística de tomada de decisão prévia no processo de codificação, orientada ao projeto de *hardware*, para uma determinação eficiente do custo **J** descrito em (4). A arquitetura implementada, como poderá ser visto no próximo capítulo, fornece as informações suficientes para apoiar os processos de decisão e de codificação.

4 ARQUITETURA PARA PREDIÇÃO INTRA-QUADRO DO PADRÃO H.264/AVC

Este capítulo trata da proposta e desenvolvimento de uma arquitetura de *hardware* específica para a predição intra-quadro que compõe o codificador de vídeo do padrão H.264/AVC. Esta arquitetura foi desenvolvida com o objetivo de atingir alto desempenho, de forma a possibilitar a codificação em tempo-real para vídeos de alta definição. As seguintes seções detalharão a arquitetura desenvolvida, a integração com os módulos de transformadas e quantização diretas e inversas desenvolvidos em conjunto com o grupo de pesquisa, os resultados de síntese para plataformas FPGA e ASIC e a metodologia utilizada para verificação.

4.1 Investigação Arquitetural para Predição Intra-Quadro

A arquitetura para predição intra-quadro partiu da definição dos seguintes requisitos:

- Processar vídeos com resolução 1080p (1920x1088), com amostragem 4:2:0, em tempo-real, a 30 quadros por segundo, o que resulta no processamento de 244800 macroblocos/segundo;
- Ocupar a menor área em *hardware* para atingir o desempenho requerido;
- Ser compatível com o padrão H.264/AVC no perfil *Main*.

Para atingir o desempenho requerido, foi realizada uma investigação sobre arquiteturas que cumprissem estes requisitos. Buscou-se então explorar o paralelismo do *hardware* de forma a aumentar a vazão de dados de saída (*throughput*). Desta forma, a primeira solução arquitetural para a predição intra-quadro foi utilizar dois caminhos de dados (*datapath*). O primeiro caminho de dados calcula a predição, SAD e resíduos para todos os modos Intra 4x4 em paralelo. O segundo caminho de dados realiza o processamento para todos os modos Intra 16x16 em paralelo com todos os modos de crominância, como mostrado na Figura 4.1. Esta primeira solução arquitetural foi baseada no trabalho de Suh, Park e Cho (SUH, 2005), sendo logo descartada quando percebido que os módulos de predição, cálculo do SAD e decisão ficariam ociosos quando os módulos T/Q e TI/QI estiverem em funcionamento, devido à dependência de dados que forma o laço de codificação, como foi mostrado no capítulo 3.

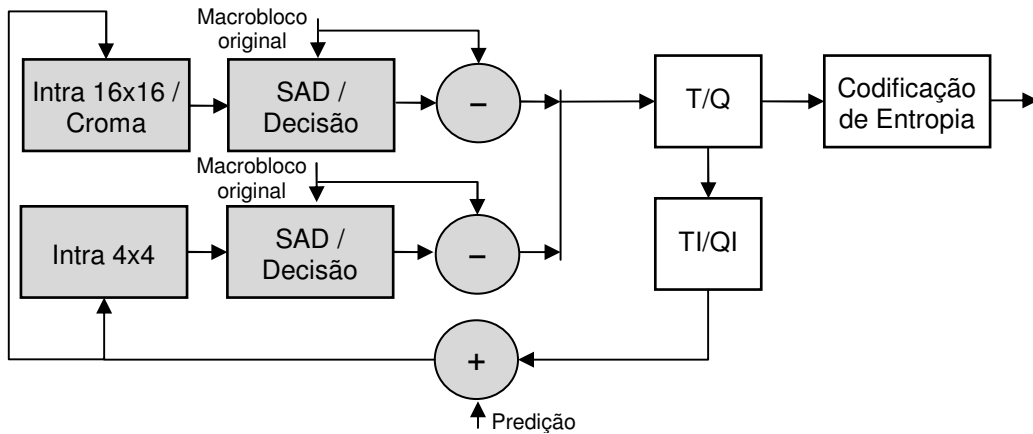


Figura 4.1: Primeira solução arquitetural para a predição intra-quadro.

A segunda solução arquitetural para a predição intra-quadro é mostrada na Figura 4.2. Nesta solução há somente um caminho de dados para predição, cálculo do SAD e decisão e o processo é multiplexado no tempo para os modos Intra 16x16 e Intra 4x4. Este esquema de tempo compartilhado foi proposto por Huang et al. (HUANG, 2005) para solucionar o problema de dependência da predição Intra 4x4, e desta forma usar menor área em *hardware* para o processo, em comparação com a primeira solução. A vantagem do esquema de tempo compartilhado é ilustrada na Figura 4.3.

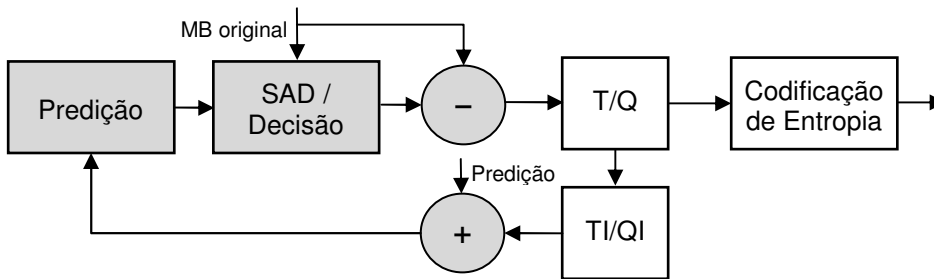


Figura 4.2: Segunda solução arquitetural para a predição intra-quadro.

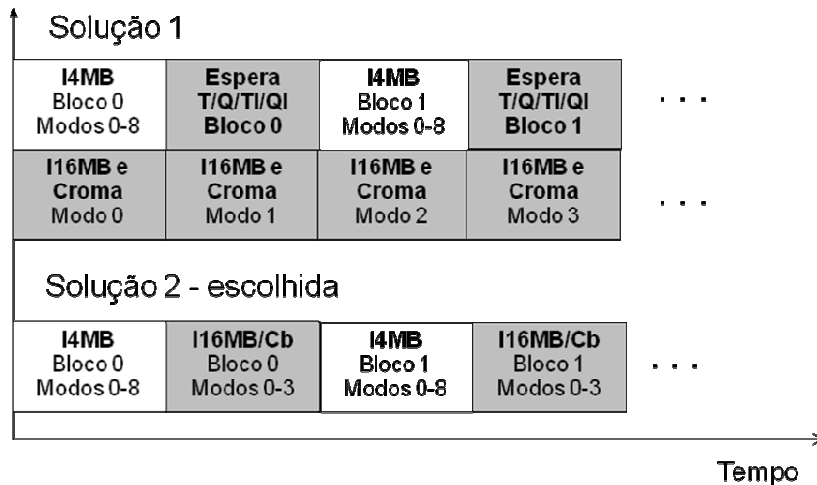


Figura 4.3: Diagrama de tempo comparando as duas soluções arquiteturais.

Na Figura 4.3, a primeira solução resulta em “bolhas” no processo de codificação, representando o tempo que os preditores esperam pelas amostras reconstruídas do laço de codificação T/Q/TI/QI. Na solução 2, os modos Intra 16x16 (I16MB) e de crominância (Cb e Cr) são particionados também em blocos 4x4 e processados nos intervalos do processamento do Intra 4x4 (I4MB). Esta solução resulta em *buffers* adicionais para armazenamento de resultados intermediários, mas economiza um módulo inteiro de cálculo de SAD e decisão com a técnica de multiplexação no tempo. Esta solução foi escolhida para o desenvolvimento da arquitetura de predição intra-quadro, que será detalhada na próxima seção.

4.2 Arquitetura para Predição Intra-Quadro

Baseada na investigação de arquiteturas detalhada anteriormente, uma arquitetura de *hardware* para a predição intra-quadro foi proposta e desenvolvida em VHDL (*VHSIC Hardware Description Language*). O trabalho de Zatt et al. (ZATT, 2008), deste grupo de pesquisa, utilizou esta mesma arquitetura de predição intra-quadro para compor o modelo de um codificador H.264/AVC completo em SystemC. A arquitetura proposta para predição intra-quadro é formada pelos principais módulos, organizados em *pipeline*:

- Gerenciamento das amostras vizinhas e máquina de controle: mantém uma memória local para armazenamento das amostras vizinhas reconstruídas, usadas para a predição, e realiza o controle global do preditor intra-quadro;
- Preditor de amostras: realiza os cálculos de predição intra-quadro para todos os modos de luminância e crominância, gerando um bloco 4x4 predito;
- Cálculo do SAD e decisão: faz o cálculo do SAD entre os blocos 4x4 predito e original e realiza uma decisão local (somente entre os modos I4MB, entre os modos I16MB e entre os modos Croma);
- *Buffer* de predição: guarda as amostras preditas para cálculo dos resíduos, reconstrução e armazenamento até a tomada de decisão I16MB/Croma, sendo composto de um conjunto de registradores (para predição I4MB) e de uma memória RAM (para I16MB/Croma);
- Cálculo de resíduos (sinal de subtração na Figura 4.4) e reconstrução (sinal de soma na Figura 4.4), sendo que a reconstrução é a soma da predição escolhida com a saída do bloco TI/QI.

A Figura 4.4 mostra um diagrama da arquitetura intra-quadro proposta e sua relação com demais módulos que compõem um codificador de vídeo básico, que gera somente quadros do tipo Intra.

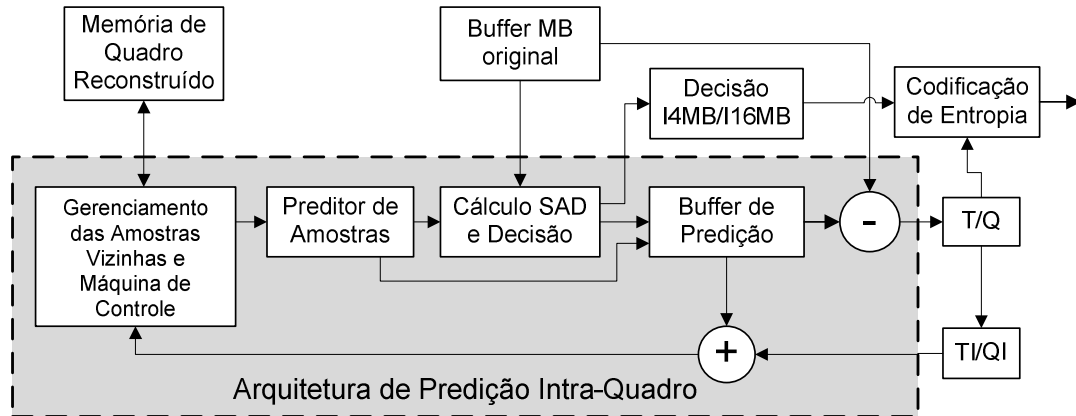


Figura 4.4: Arquitetura proposta e desenvolvida a predição intra-quadro e sua relação com os demais módulos que compõem um codificador de vídeo somente para quadros Intra.

Os módulos externos à arquitetura não fazem parte do foco principal deste trabalho e, por este motivo, suas arquiteturas internas não serão tratadas nesta seção. A seção 4.3 mostra somente o processo de integração da arquitetura com os módulos T/Q e TI/QI, desenvolvidos em conjunto com o grupo de pesquisa. O desenvolvimento dos demais módulos é importante para integração de um codificador de vídeo básico, porém, por sua complexidade de desenvolvimento e para delimitar o escopo do trabalho eles não serão tratados no texto. Alguns módulos externos à arquitetura já estão em desenvolvimento e a integração dos módulos à arquitetura desenvolvida neste trabalho também já está prevista como trabalho futuro do grupo de pesquisa.

As seções seguintes descreverão em detalhes os módulos que compõem a arquitetura de predição intra-quadro.

4.2.1 Gerenciamento das Amostras Vizinhas e Máquina de Controle

Este módulo realiza o gerenciamento das amostras vizinhas reconstruídas de luminância (Y) e croma (Cb e Cr), que serão usadas como entrada para o preditor de amostras. Uma memória RAM (*Random Access Memory*) interna é usada para armazenar as amostras reconstruídas para predição de uma linha de macroblocos dentro de um quadro. Esta estratégia é adotada para reduzir a largura de banda de memória externa ao *chip* necessária para armazenar o quadro reconstruído, do contrário, para cada processamento de macrobloco as amostras reconstruídas das bordas de cada macrobloco deveriam ser salvas na memória externa, consumindo ciclos extras neste processo. As amostras vizinhas da esquerda do macrobloco são armazenadas em registradores.

A Figura 4.5 mostra um exemplo do conteúdo da memória em um instante de tempo (blocos em cinza na Figura 4.5) para um quadro QCIF, que contém 11 macroblocos em uma linha. A memória armazena as amostras vizinhas dos macroblocos 0-7 que já foram processados e serão usadas como referência para predição intra-quadro dos macroblocos que seguem o macrobloco corrente (MB corrente) na ordem de codificação de macroblocos (*raster*). Os macroblocos 8-11, processados anteriormente ao MB corrente, já usaram amostras de macroblocos que antecederam o macrobloco 0 na ordem de codificação e, portanto, as amostras de blocos anteriores ao macrobloco 0 não são mais

usadas no processo de codificação. Ou seja, a medida que cada macrobloco é processado pela predição intra-quadro e reconstruído, o conteúdo da memória que guarda as amostras da linha superior usada como referência é substituída pelas amostras processadas pela linha inferior, que servirá de referência para a próxima linha (linha abaixo) de macroblocos, assim por diante até o final do processamento do quadro. Esta memória foi dimensionada para guardar uma linha de amostras de 8 bits de um quadro do vídeo com resolução 1080p, que é a máxima resolução suportada, ou seja, 1920 amostras de luminância e 960 amostras para cada canal de crominância. O tamanho total de memória usado é de 30720 bits. Para suportar resoluções maiores, é trivial a alteração do tamanho desta memória por um parâmetro do módulo VHDL.

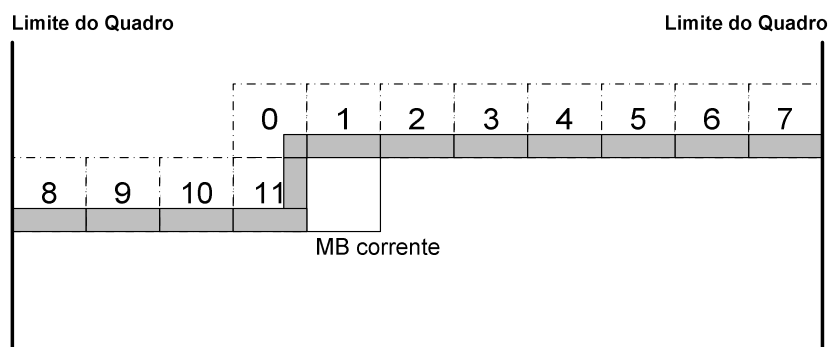


Figura 4.5: Amostras reconstruídas (em cinza) armazenadas em memória interna para predição.

Além do gerenciamento da memória, este módulo realiza o controle do processo de multiplexação no tempo mostrado na Figura 4.3 e também gera o valor DC do modo Intra 16x16 e os parâmetros **H**, **V**, **a**, **b** e **c** do modo Plano (Intra 16x16), cujas equações foram descritas no capítulo 3. A geração destes parâmetros é feita antes de começar o processo de predição para cada macrobloco. Uma FSM (*Finite State Machine*) com 44 estados foi descrita para realizar o processo de leitura da memória e geração dos parâmetros. Outra FSM, com 12 estados, foi descrita para armazenar os resultados novamente na memória de amostras vizinhas. As Figuras 4.6a e 4.6b mostram os diagramas das duas FSM. Por simplificação, os estados de ambas as máquinas foram representados em macro-estados de acordo com sua funcionalidade.

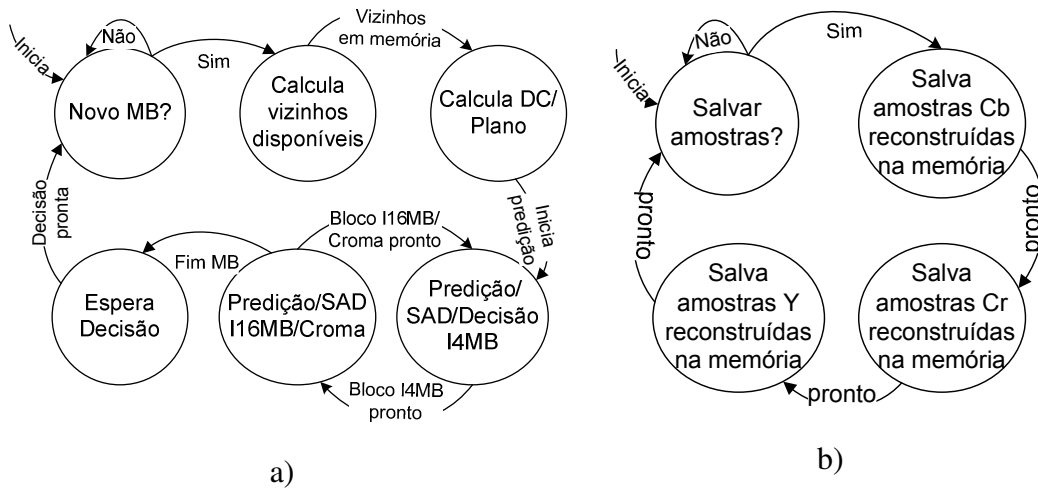


Figura 4.6: Diagrama das máquinas de controle. a) FSM de controle e leitura das amostras vizinhas; b) FSM de armazenamento das amostras vizinhas.

4.2.2 Preditor de Amostras

O preditor de amostras é o módulo que realiza as operações da predição intra-quadro definidas no H.264/AVC. Este módulo tem como entrada as amostras reconstruídas (já processadas anteriormente) da máquina de gerenciamento de vizinhos. Ele opera em dois regimes: i) gerar 4 amostras preditas por ciclo para todos os 9 modos Intra 4x4 em paralelo; ii) gerar 4 amostras preditas por ciclo para os 4 modos Intra 16x16 e os 4 modos de crominância em paralelo. Em ambos os regimes, a predição de um bloco 4x4 é concluída após 4 ciclos para todos modos de predição em paralelo.

4.2.2.1 Modos de Predição Intra 4x4

Os modos horizontais e verticais, não representam custo em *hardware* em termos de operadores, pois o bloco predito é gerado pela cópia das amostras vizinhas já reconstruídas. O modo DC e os modos diagonais do Intra 4x4 (modos 3 a 8) devem ser calculados de acordo com o que foi mostrado na Tabela 3.1. Observando a Tabela 3.1, podem ser extraídos três padrões de equações, (36) (37) e (38), para calcular as amostras preditas de todos os modos diagonais.

$$S_1 = (A + B + 1) \gg 1 \quad (36)$$

$$S_2 = (A + 2 \cdot B + C + 2) \gg 2 \quad (37)$$

$$S_3 = (A + 3 \cdot B + 2) \gg 2 \quad (38)$$

Observa-se ainda que existe muito compartilhamento de sub-operações para um mesmo modo ou entre diferentes modos de predição, seja ela do tipo S_1 , S_2 ou S_3 , com as mesmas amostras de entrada. A Tabela 4.1 mostra somente os cálculos necessários para gerar todas as amostras preditas (de P_0 a P_{22}) de todos os modos diagonais.

Tabela 4.1: Operações Necessárias para Cálculo dos Modos de Diagonais (I4MB).

Equação	L	K	J	I	M	A	B	C	D	E	F	G	H	Soma	Desloca	Amostra predita
(36)	1	1												1	>> 1	P ₀
		1	1											1	>> 1	P ₁
			1	1										1	>> 1	P ₂
				1	1									1	>> 1	P ₃
					1	1								1	>> 1	P ₄
						1	1							1	>> 1	P ₅
							1	1						1	>> 1	P ₆
								1	1					1	>> 1	P ₇
									1	1				1	>> 1	P ₈
(37)	1	2	1											2	>> 2	P ₁₀
		1	2	1										2	>> 2	P ₁₁
			1	2	1									2	>> 2	P ₁₂
				1	2	1								2	>> 2	P ₁₃
					1	2	1							2	>> 2	P ₁₄
						1	2	1						2	>> 2	P ₁₅
							1	2	1					2	>> 2	P ₁₆
								1	2	1				2	>> 2	P ₁₇
									1	2	1			2	>> 2	P ₁₈
										1	2	1		2	>> 2	P ₁₉
											1	2	1	2	>> 2	P ₂₀
(38)	3	1												2	>> 2	P ₂₁
												1	3	2	>> 2	P ₂₂

Assim, a primeira solução adotada para o cálculo dos modos diagonais foi a implementação de três diferentes tipos de módulos, um para cada equação (36), (37) e (38), como é mostrado na Figura 4.7. As equações (37) e (38) foram reorganizadas de forma a diminuir o número de operadores. Para gerar todas as amostras, são necessários dez módulos do tipo 1, onze módulos do tipo 2 e dois módulos do tipo 3, totalizando 36 somadores. Os deslocamentos e somas de '1' não representam custo adicional de área de *hardware* (o bit '1' pode ser colocado na entrada *carry-in* de um somador completo). Todas as operações são feitas em paralelo, em 1 ciclo de relógio.

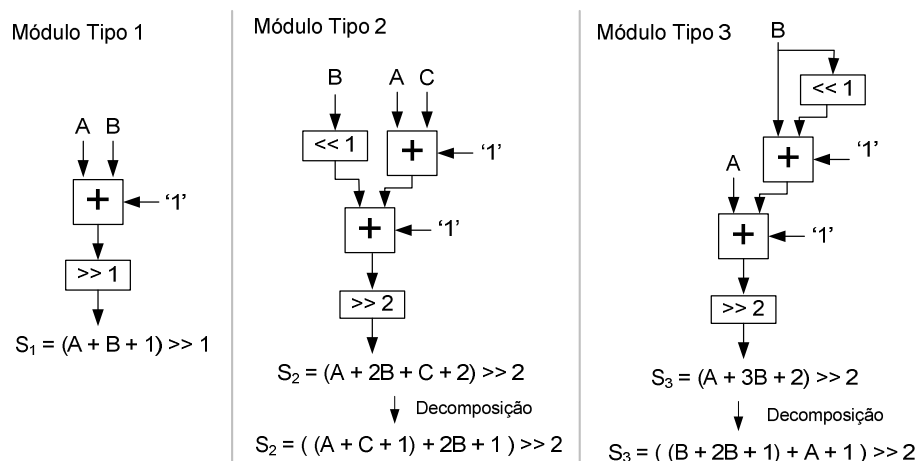


Figura 4.7: Arquitetura do preditor para modos diagonais (I4MB) usando decomposição de equações.

A segunda solução adotada, baseada no trabalho de Lin (LIN, 2007), utiliza a eliminação de sub-expressões comuns nas equações (37) e (38), pois muitas delas coincidem com os resultados das equações do tipo (36). A solução é mostrada na Figura 4.8. Percebe-se que ela utiliza mais somadores para o cálculo de cada amostra, em comparação com a primeira solução. Entretanto, se verificado no total, somente 25 somadores são necessários para cobrir todas as possibilidades de cálculo dos modos diagonais. Esta abordagem não insere acréscimo no caminho crítico (2 somadores), o que não afeta o desempenho da arquitetura. Vale ressaltar que o trabalho de Lin (LIN, 2007) não detalha a arquitetura interna do preditor, como é feito neste trabalho. Ele somente mostra que com esta decomposição é possível reduzir operadores na predição dos modos diagonais.

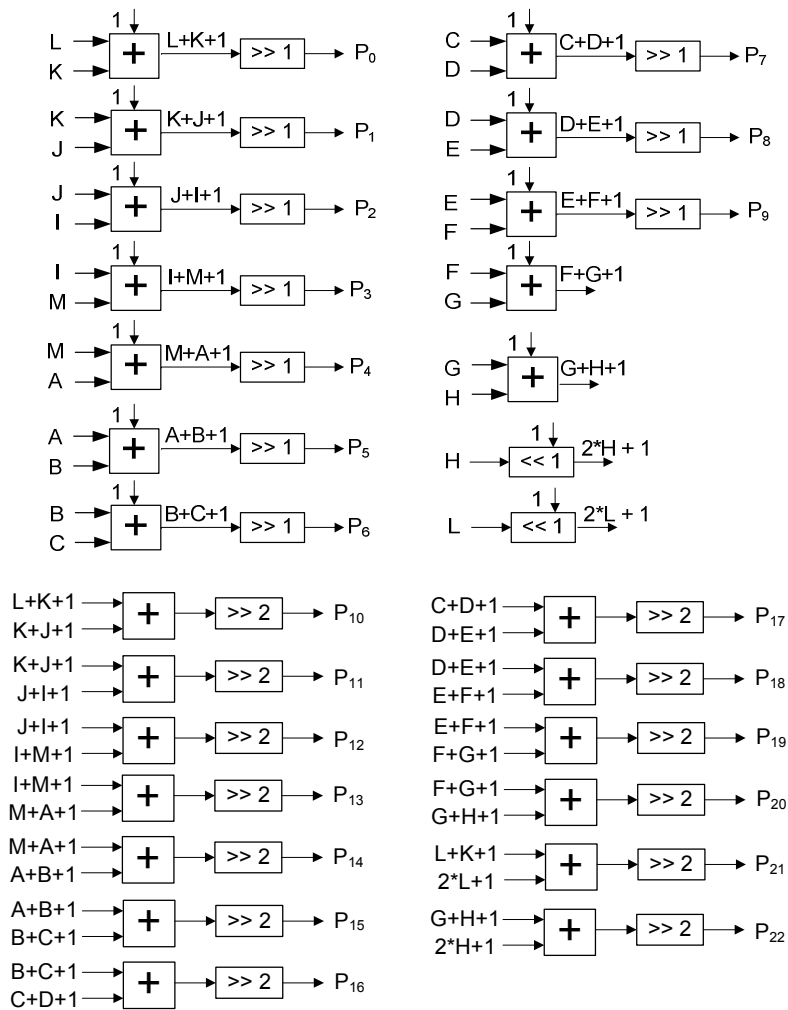


Figura 4.8: Arquitetura do preditor para modos diagonais (I4MB) usando eliminação de sub-expressões comuns.

Uma adaptação do esquema mostrado na Figura 4.8 foi utilizada para o cálculo do modo DC I4MB. Ele é feito pela adição de mais 3 somadores, compartilhando o resultado das operações do primeiro estágio de cálculo dos modos diagonais ($L+K+1$, $J+I+1$, $A+B+1$ e $C+D+1$). Os somadores adicionais não afetaram o caminho crítico,

pois dois deles são colocados em paralelo com o segundo estágio de cálculo mostrado na Figura 4.8.

A Tabela 4.2 resume o ganho de área (em número de somadores) obtido pela segunda solução em relação à primeira solução.

Tabela 4.2: Comparação Entre Soluções para Cálculo dos Modos Diagonais (I4MB).

Solução Arquitetural	Número de somadores	Redução
Decomposição de equações (Figura 4.7)	36	-
Eliminação de sub-expressões comuns (Figura 4.8)	25	30,5%

4.2.2.2 Modos de Predição Intra 16x16 e Crominância

Os modos Intra 16x16 e de crominância são similares, por isto são tratados na mesma seção.

Assim como no tipo Intra 4x4, os modos horizontal e vertical do Intra 16x16 e de crominância não representam custo de *hardware*, pois são simples cópias das amostras reconstruídas. O modo DC (I16MB) é calculado em 7 ciclos pela FSM de gerenciamento das amostras vizinhas, tratada na seção 4.2.1. Os modos planos do I16MB e de crominância são em parte calculados pela FSM, que gera os parâmetros **a**, **b** e **c**. O preditor de amostras implementa somente as operações (24) e (30).

A Figura 4.9 mostra a parte da operação do modo plano que está incluída no preditor de amostras. Ela gera as 4 amostras preditas/ciclo do modo plano, onde **x** e **y** são os índices da amostra e **a**, **b** e **c** são os parâmetros, todos estes fornecidos pela FSM de gerenciamento das amostras vizinhas. A operação de soma das amostras **a** e **c** e o arredondamento é feita em um ciclo, para cada uma das 4 linhas do bloco 4x4. As demais operações que geram as 4 amostras são feitas no próximo ciclo, também para cada linha, em paralelo com a operação de soma **a** e **c** e o arredondamento da próxima linha de 4 amostras, pois não há dependência de dados neste caso. Este módulo é replicado para gerar as amostras de crominância no modo plano em paralelo. São utilizados 10 multiplicadores, no total, para cálculo do modo plano para luminância (Intra 16x16) e crominância.

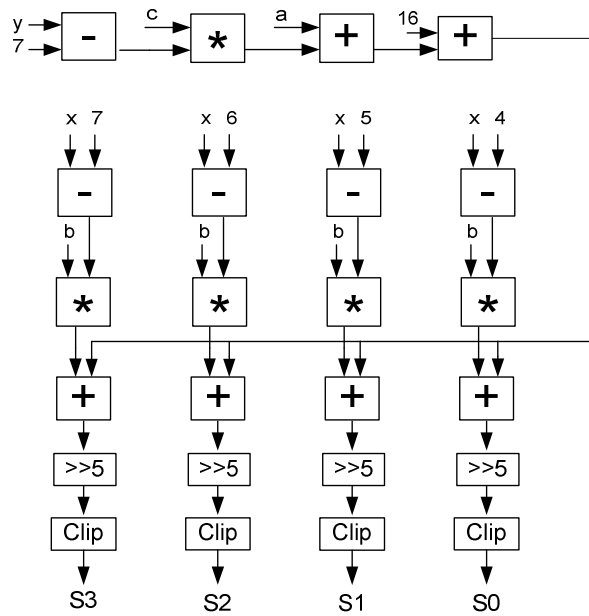


Figura 4.9: Modo Plano I16MB Usando Multiplicadores.

As multiplicações na Figura 4.9 são feitas entre os parâmetros b e c e os índices x e y , que só dependem da posição da amostra no bloco, portanto, são constantes. Desta forma, uma versão sem multiplicadores foi desenvolvida com o propósito de reduzir área de *hardware*. Esta solução usa somente somadores, multiplexadores e deslocamentos, como é mostrado na Figura 4.10.

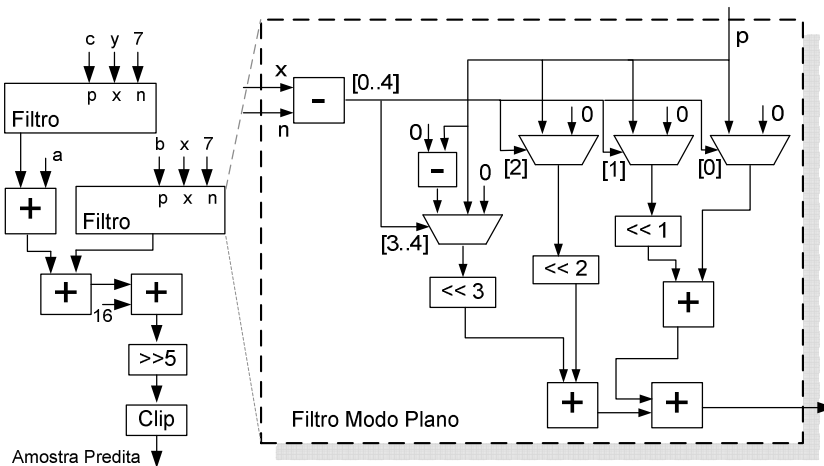


Figura 4.10: Modo Plano I16MB Usando Apenas Somas e Deslocamentos.

4.2.2.3 Controle do Preditor de Amostras

Uma máquina de estados (FSM), com 8 estados, foi desenvolvida para controlar os multiplexadores que selecionam somente as 4 amostras preditas que são propagadas para a saída a cada instante de tempo. Estas amostras são armazenadas no *buffer* de

predição para posterior uso pelos módulos de cálculo do SAD, cálculo de resíduos e reconstrução do macrobloco.

4.2.3 Cálculo do SAD

O módulo de cálculo do SAD está mostrado na Figura 4.11. Este módulo é dividido em 9 elementos de processamento (EP0 a EP8). Cada EP calcula a diferença entre as amostras do bloco 4x4 original (*Orig*), a ser codificado, e o bloco 4x4 predito (*Pred*), saída do preditor de amostras, acumulando o valor absoluto desta diferença. O módulo faz a leitura de 4 amostras por ciclo para 9 modos de predição em paralelo.

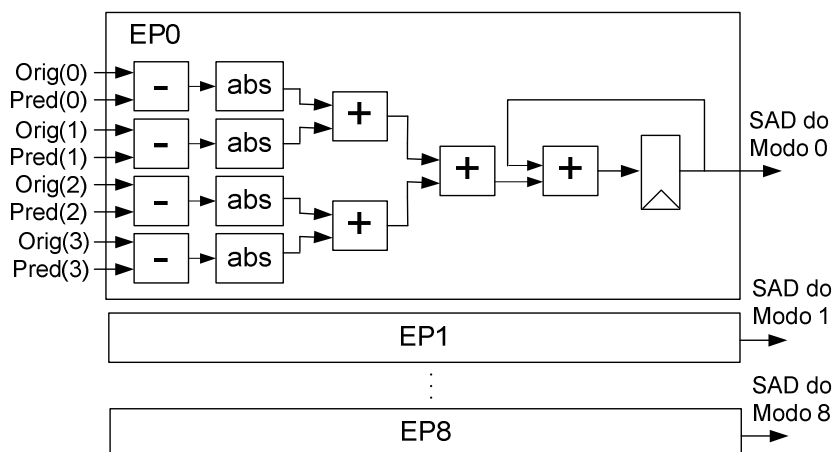


Figura 4.11: Arquitetura para cálculo do SAD.

Cada EP é dividido em 3 estágios de *pipeline*, porém as barreiras de *pipeline* estão omitidas na Figura 4.11. O primeiro estágio contém 4 subtratores e 4 módulos de cálculo de valor absoluto, para calcular a diferença absoluta para quatro amostras. O segundo estágio contém uma árvore de somadores, que acumula as diferenças para as 4 amostras. O último estágio é formado por um registrador, que guarda as diferenças acumuladas da linha de 4 amostras processada anteriormente, e o somador que adiciona este valor com o resultado do SAD para a linha de 4 amostras atual. Este conjunto de somadores e registradores é usado para calcular o SAD para os 9 modos I4MB.

O cálculo de SAD para os modos I16MB e crominância utilizam o mesmo caminho de dados do I4MB, mostrado na Figura 4.11. Neste caso, os valores SAD dos 4 modos I16MB usam os EP0 a EP3, e os valores de SAD para crominância usam os EP4 a EP7. Outro conjunto de somadores e registradores é usado para guardar os resultados de SAD para todos os modos I16MB e crominância, pois estes valores serão usados somente no final do processamento do macrobloco, onde eles serão consultados para a tomada de decisão do melhor modo.

O módulo de cálculo de SAD trabalha em *pipeline* com o módulo de predição: assim que as 4 amostras foram preditas, o módulo de cálculo do SAD dispara a árvore de somadores, e o módulo preditor passa a calcular as próximas 4 amostras, dependendo do modo de predição (I4MB ou I16MB). Desta forma, após 5 ciclos o cálculo do SAD de um bloco 4x4 é concluído para todos os modos em paralelo.

4.2.4 Decisão do Modo de Predição

O etapa de decisão do modo de predição é dividido em 3 módulos: i) Decisão do modo I4MB; ii) Decisão do modo I16MB; iii) Decisão do modo cromina. O módulo de decisão I4MB recebe como entrada os valores de SAD dos 9 modos I4MB em paralelo, bem como a informação da disponibilidade do modo, calculada com base na disponibilidade dos vizinhos, que é fornecida pela FSM de gerenciamento dos vizinhos. A Figura 4.12 mostra a arquitetura do módulo de decisão de modo I4MB. Este módulo realiza a escolha entre os 9 modos I4MB com base no menor valor de SAD. O módulo é implementado como uma árvore de comparadores, que tem como entrada o SAD de cada modo e como saída o SAD do modo escolhido. Este módulo fornece ainda como saída o índice do modo I4MB. A decisão do modo I4MB é feita em 3 ciclos (embora as barreiras de *pipeline* estejam omitidas na Figura 4.12) após o cálculo do SAD para todo bloco 4x4.

A Figura 4.13 mostra a arquitetura do módulo de decisão de modo I16MB. Este módulo escolhe o modo que produz o menor valor de SAD dentre os 4 modos I16MB. A implementação deste módulo é realizada da mesma maneira que a decisão I4MB, como uma árvore de comparadores que tem como entrada o SAD de cada modo e como saída o SAD do modo escolhido. No caso da predição I16MB, a decisão é feita somente após o cálculo do SAD para todas as amostras previstas do macrobloco. Não é possível tomar alguma decisão prévia do menor valor de SAD até que este seja calculado para todas as amostras do macrobloco, para todos os modos I16MB. Para isto as amostras previstas são armazenadas em uma memória de predição, que será abordada na próxima seção.

A Figura 4.14 mostra a arquitetura do módulo de decisão do modo de crominância. A escolha do modo de crominância também deve ser tomada após o cálculo de SAD para todas as amostras de crominância previstas de um macrobloco. A decisão também é feita pelo menor SAD dentre os 4 modos de predição. Além disto, como só um modo de predição é escolhido para os dois canais de crominância, Cb e Cr, nesta arquitetura o modo com menor valor de SAD dentre Cb e Cr é escolhido como melhor modo de cromina.

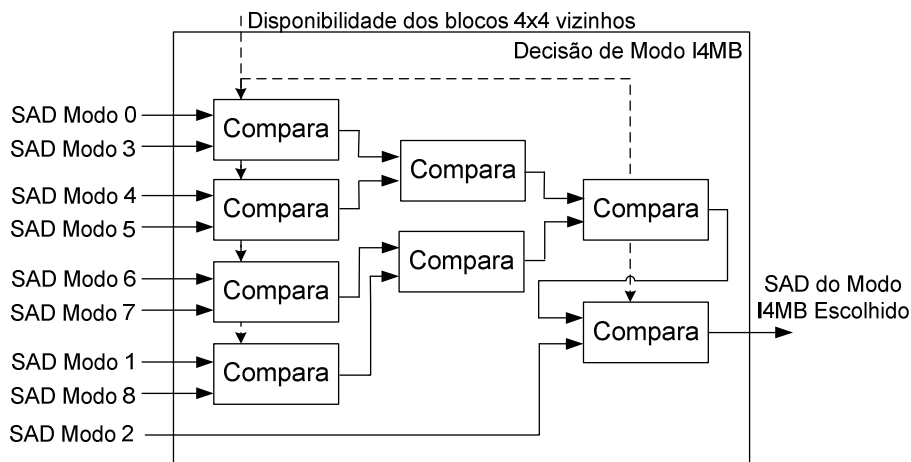


Figura 4.12: Arquitetura para Decisão do Modo de Predição I4MB.

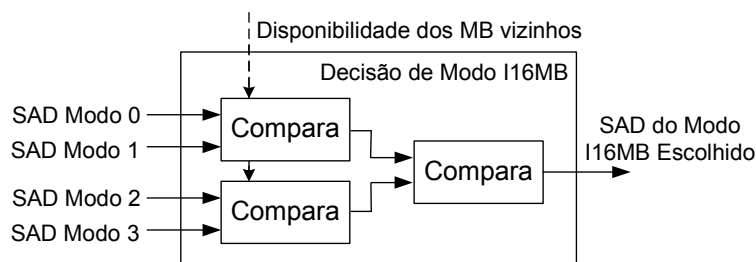


Figura 4.13: Arquitetura para Decisão do Modo de Predição I16MB.

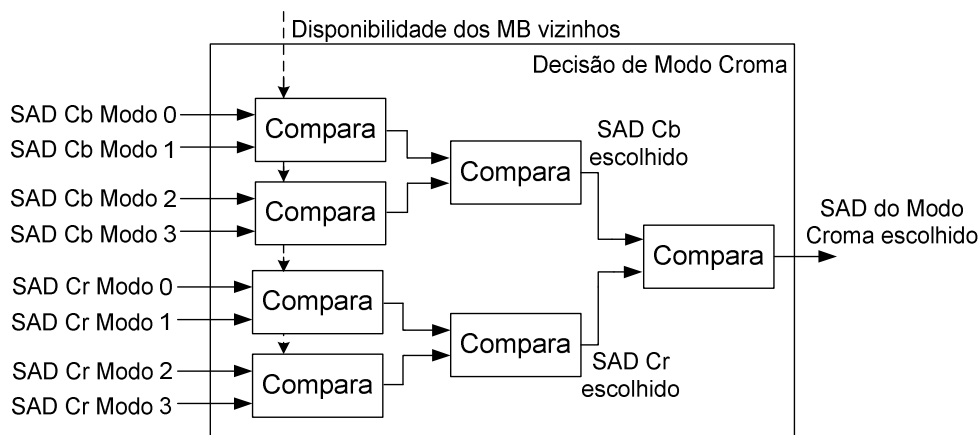


Figura 4.14: Arquitetura para Decisão do Modo de Predição Croma.

A arquitetura de predição intra-quadro faz a escolha dentre os modos I4MB para cada bloco 4x4 de um macrobloco, dentre os modos Intra 16x16 e os modos de crominância. Como foi mencionado no capítulo 3, este trabalho trata somente destas decisões. A escolha do melhor tipo de macrobloco a ser usado (Intra 4x4 ou Intra 16x16), respeitando uma relação taxa-distorção, é deixada a cargo do controle do codificador, não sendo abordado neste trabalho.

4.2.5 Buffer e Memória de Predição

As amostras previstas do modo de predição selecionado na etapa de decisão são usadas tanto para geração dos resíduos, entrada do bloco de transformadas, quanto na reconstrução, que consiste na soma da predição escolhida com os resíduos de saída do bloco de transformadas inversas. Devido ao processamento em *pipeline* das etapas de predição, cálculo de SAD e decisão do modo, no momento de geração dos resíduos e reconstrução as amostras previstas a serem processadas não estão mais disponíveis. Para obter as amostras nestes dois momentos, duas abordagens podem ser usadas:

- Armazenar as amostras previstas em *buffers* ou memória. Deve ser armazenado um bloco de 4x4 amostras para todas os modos, no caso da predição I4MB (pois é tomada uma decisão local para cada bloco 4x4), e um macrobloco (16x16 amostras no caso de luminância e 8x8 amostras para crominância) para todos os modos, até que a decisão I16MB e Croma seja tomada;
- Utilizar uma unidade de recálculo da predição.

A primeira abordagem utiliza maior área de *hardware* (em memória), mas economiza ciclos de processamento em comparação com a segunda abordagem. Como o desenvolvimento arquitetural foi guiado com o objetivo de atingir desempenho para codificar vídeos HDTV, a primeira abordagem foi escolhida, a fim de diminuir o número de ciclos necessários para concluir a codificação de um macrobloco. Além disso, apesar da memória RAM ser um recurso caro no caso da síntese para ASICs, para o caso de mapeamento da arquitetura para alguns dispositivos FPGA, o recurso de memória RAM já está disponível na forma de block RAMs (BRAM), como é o caso dos dispositivos FPGA utilizados para síntese neste trabalho.

As amostras preditas dos 9 modos Intra 4x4 são armazenadas em um *buffer*, formado por registradores, até que a decisão do modo de predição seja realizada. Quando tal decisão é concluída, somente as amostras preditas do modo escolhido são selecionadas para o cálculo de resíduos e as demais amostras dos modos não selecionados são descartadas. Este *buffer* é composto por 144 registradores de 8 bits para armazenar todas as 16 amostras (bloco de 4x4 amostras) para os 9 modos de predição. Um *buffer* adicional é usado para armazenar as 16 amostras do modo Intra 4x4 escolhido a ser usado na reconstrução do bloco 4x4, ou seja, 16 registradores de 8 bits são usados neste caso. Ao final da codificação, caso a seleção seja feita pelo modo Intra 4x4, somente as amostras preditas nos modos selecionados devem ser enviados à etapa de transformadas e quantização. Porém, como este processo já é realizado antes de tomada a decisão final (usar particionamento Intra 4x4 ou Intra 16x16), e uma escolha prévia é feita para cada bloco 4x4, uma abordagem mais interessante é armazenar em memória os coeficientes resultantes da etapa de quantização que passaram pela predição Intra 4x4. Assim, caso a predição Intra 4x4 seja selecionada, os coeficientes estão prontos para serem enviados à codificação de entropia.

As amostras preditas nos modos Intra 16x16 e nos modos de crominância (Cb e Cr), de todo o macrobloco, são armazenadas em duas memórias RAM distintas, até que a decisão do modo de predição do macrobloco (I4MB/I16MB, não tratada neste trabalho) seja realizada ao final. Duas memórias RAM *single-port* foram utilizadas, uma delas para as amostras de luminância no modo Intra 16x16, e a outra para as amostras de crominância Cb e Cr. As memórias guardam todas as amostras preditas nos modos Intra 16x6, Cb e Cr de um macrobloco. Um macrobloco contém 16 blocos (0 a 15) de 16 amostras de luminância (bloco 4x4) e 8 blocos de 16 amostras de crominância. Considerando amostras de 8 bits, a memória de luminância guarda 8192 bits e a memória de crominância guarda 4096 bits. No total, são 12288 bits para a memória de predição.

A organização das memórias é mostrada na Figura 4.15. Esta organização facilita com que as 16 amostras do modo de predição escolhido estejam disponíveis em um ciclo de leitura para o módulo de cálculo dos resíduos. A Figura 4.15 mostra o exemplo de memória para luminância, onde cada linha armazena um bloco 4x4 (16 amostras) predito utilizando um modo de predição (horizontal, vertical, DC ou plano). A memória é organizada em 64 linhas de 128 bits. A memória de predição para crominância possui somente 8 blocos, 4 para Cb e 4 para Cr, sendo organizada em 32 linhas de 128 bits. A memória é dividida em duas, sendo 16 linhas para Cb e 16 linhas para Cr.

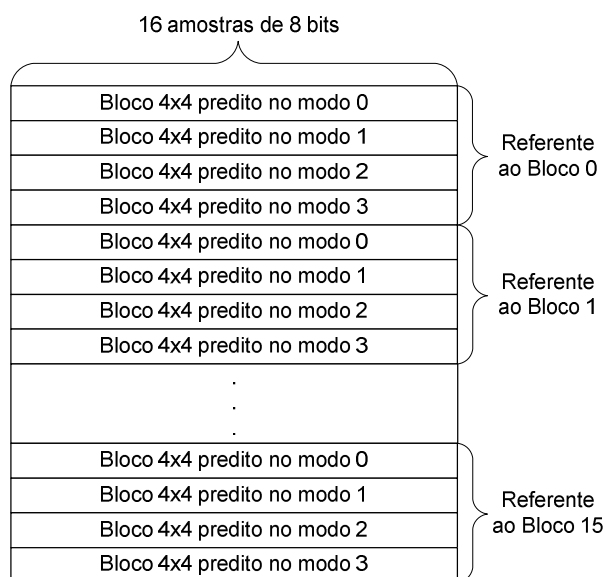


Figura 4.15: Organização das Memórias de Predição para Intra 16x16 e Crominância.

Como a organização das memórias não é compatível com a ordem com que são geradas as amostras preditas (4 amostras por ciclo, para todos os modos em paralelo), o mesmo *buffer* que armazena as amostras do modo Intra 4x4 é reusado em outro momento para armazenar as amostras preditas nos modos Intra 16x16 e crominância, antes da escrita na memória de predição. O *buffer* permite que as amostras sejam reorganizadas para escrita de todas as 16 amostras na memória em 1 ciclo de relógio. Duas máquinas de estado (FSM) foram descritas para a escrita e leitura dos dados corretos nas memórias de predição.

4.2.6 Cálculo de Resíduos e Reconstrução

O cálculo de resíduos consiste na subtração da amostra correspondente do bloco a ser codificado, de luminância ou crominância, com a amostra predita em um modo de predição escolhido. O módulo de cálculo dos resíduos contém 16 subtratores de 8 bits ligados a 16 multiplexadores. Cada multiplexador tem como entrada uma amostra predita para cada modo escolhido dos tipos Intra 4x4, Intra 16x16 ou Crominância. A outra entrada do subtrator contém um multiplexador para selecionar a amostra de luminância ou crominância correspondente no bloco original, a ser codificado. A saída de um subtrator representa o resíduo de predição e serve como entrada para o módulo de transformadas diretas. A operação de cálculo do resíduo é realizada em 1 ciclo para todas as 16 amostras em paralelo.

A reconstrução consiste na soma da predição escolhida com os resíduos já transformados e quantizados, provenientes da etapa de transformadas inversas. O módulo de reconstrução é formado por 16 somadores que tem como entrada os 16 coeficientes resultantes do bloco de transformadas inversas e um multiplexador, semelhante ao do cálculo dos resíduos, que seleciona o tipo da amostra, seja Intra 4x4, Intra 16x16 ou crominância. Este cálculo é feito também em um ciclo de relógio.

4.3 Integração com os Módulos de Transformadas e Quantizações Diretas e Inversas

Como foi tratado na seção 4.1, o projeto da arquitetura para a predição intra-quadro do padrão H.264/AVC tem relação direta com o projeto dos módulos de transformadas e quantizações diretas e inversas, devido à dependência de dados entre amostras de blocos 4x4 vizinhos e entre amostras de macroblocos vizinhos. O processamento do tipo Intra 4x4, particularmente, limita o desempenho das arquiteturas, pois insere latência no processo de codificação do macrobloco Intra. Isto dificulta a tradicional divisão em macro-estágios de *pipeline* da etapa de busca da predição e a etapa de codificação propriamente dita, que envolve as transformadas e quantizações.

Com isto, foi desenvolvido um trabalho em parceria com o Grupo de Arquiteturas e Circuitos Integrados (GACI), da Universidade Federal de Pelotas (UFPEL), com o intuito de projetar e desenvolver arquiteturas com baixa latência e alta vazão de dados de saída (*throughput*) para os módulos de transformadas diretas e inversas do padrão H.264/AVC, visando a codificação intra-quadro. O grupo atingiu o objetivo definido inicialmente, desenvolvendo os módulos de transformadas DCT 4x4, Hadamard 4x4 e 2x2 e quantizações diretas e inversas, integrados em um único módulo denominado Loop TQ/IQIT. Os módulos desenvolvidos possuem alta vazão de dados (*throughput*) e baixa latência, pela diminuição do número de estágios de *pipeline*. Os resultados do conjunto dos trabalhos podem ser consultados em (SAMPAIO, 2009).

Portanto, o projeto do loop TQ/IQIT não foi realizado no escopo deste trabalho, mas sim a integração deste com a arquitetura de predição intra-quadro desenvolvida. A interface entre os dois módulos está mostrada na Figura 4.16. A interface de dados dos módulos consiste nos resíduos de predição, que entram no loop, e nos resíduos pós-transformação/quantização diretas e inversas, que saem do loop. A sincronização de operação dos dois módulos é feita via *handshake*.

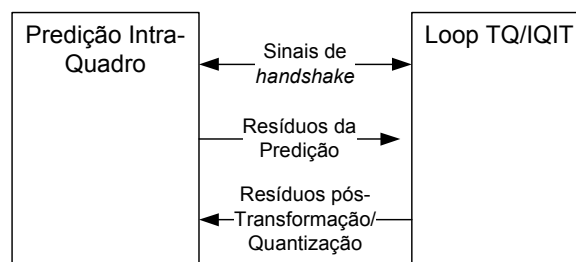


Figura 4.16: Interface Entre o Preditor Intra-Quadro e o Loop TQ/IQIT.

4.4 Resultados de Síntese

Todos os módulos da arquitetura de predição intra-quadro foram desenvolvidos em VHDL, e sintetizados para dispositivos FPGA da Xilinx (XILINX, 2009) usando o Xilinx ISE 8.2i. A Tabela 4.3 mostra os resultados de síntese da arquitetura completa e de seus principais módulos para o dispositivo da família Virtex-II Pro da Xilinx, o XC2VP30 (XILINX, 2009a). Este dispositivo foi fabricado na tecnologia CMOS 0.13 μm .

Tabela 4.3: Resultados de Síntese para FPGA Xilinx Virtex-II Pro.

	Slices	Slice Flip-Flops	4-Input LUTs	18 Kb BRAM	Multiplicadores 18x18
Gerenciamento das Amostras Vizinhas e Máquina de Controle	1663	1888	2955	3	-
Preditor de Amostras	754	357	1428	-	10
Cálculo SAD	660	428	1076	-	-
Decisão	311	118	586	-	-
Resíduos	80	-	128	-	-
Reconstrução	271	160	496	-	-
Memória de Predição	466	135	881	8	-
Arquitetura Completa	5020	4466	8866	11	10
Uso do Dispositivo	36%	16%	32%	8%	7%

Dispositivo: Xilinx Virtex-II Pro XC2VP30-7

A arquitetura completa pode atingir uma frequência de 130 MHz, sendo que a parte lógica ocupa 36% dos slices e 32% dos elementos lógicos (LUTs) do dispositivo XC2VP30. As memórias de amostras vizinhas e de predição usam 30720 e 12288 bits de memória e foram mapeadas para 3 e 8 Block RAMs (BRAM) do dispositivo, respectivamente. Apesar da BRAM do dispositivo ter capacidade para 18 Kb, o esquema de escrita e leitura dos dados, em paralelo, força com que sejam usadas mais BRAMs para estes casos, sendo este o motivo pelo qual a memória de predição, que é menor que a memória de vizinhos, usa mais BRAMs. No total, 11 BRAMs foram utilizadas, ocupando somente 8% de todas as BRAMs disponíveis no dispositivo.

Já que os dispositivos FPGA usualmente possuem multiplicadores na sua estrutura interna, os multiplicadores de cálculo do modo plano (preditor de amostras) foram implementados usando funções-macro (usando o símbolo “*” no VHDL) para que fossem mapeados aos multiplicadores do FPGA. No total, foram utilizados 10 multiplicadores para o cálculo do modo plano.

O módulo de gerenciamento das amostras vizinhas e máquina de controle foi o que ocupou a maior parcela dos elementos lógicos da arquitetura. Isto se deve ao fato de que o cálculo dos parâmetros DC 16x16 e do modo plano foram feitos internamente nesta máquina de estados (FSM). Por simplificação da descrição VHDL, as operações neste módulo foram feitas utilizando variáveis ao invés de sinais. Por causa disto, a ferramenta de síntese não conseguiu reutilizar os operadores de um estado nos outros estados da máquina, replicando os operadores para cada operação em cada estado. Esta é a explicação porque este módulo ocupou quase o dobro de LUTs que o preditor de amostras, que é o núcleo da predição intra-quadro e o segundo módulo que ocupa mais elementos lógicos, o que é esperado pelo paralelismo utilizado para a predição de amostras.

Foi realizada uma síntese para um dispositivo mais atual da Xilinx, da família Virtex-5, o XC5VLX110 (XILINX, 2009b). Este dispositivo foi fabricado na tecnologia CMOS 65 nm. Os resultados estão mostrados na Tabela 4.4.

Tabela 4.4: Resultados de Síntese para FPGA Xilinx Virtex-5.

	Slice Registers	Slice LUTs	LUT/Flop Pairs	36 Kb BRAM/FIFO	DSP48Es
Gerenciamento das Amostras Vizinhas e Máquina de Controle	1850	2430	2659	3	-
Preditor de Amostras	380	1343	1338	-	6
Cálculo SAD	428	1004	1045	-	-
Decisão	119	482	418	-	-
Resíduos	-	128	128	-	-
Reconstrução	160	480	480	-	-
Memória de Predição	150	846	891	8	-
Arquitetura Completa	4413	7516	7857	11	6
Uso do Dispositivo	6%	10%	11%	8%	9%

Dispositivo: Xilinx Virtex-5 XC5VLX110-3

A arquitetura completa, sintetizada para o dispositivo Virtex-5, atingiu uma frequência de 135 MHz e os resultados de uso dos recursos do dispositivo são melhores que os resultados para Virtex-II Pro, pois uma LUT do dispositivo da família Virtex-5 implementa uma função lógica de 6 entradas, diferente do dispositivo da família Virtex-II Pro, no qual uma LUT implementa uma função lógica de 4 entradas. Além disto, a capacidade do dispositivo XC5VLX110 é maior do que a do XC2VP30 em termos de elementos lógicos. Nota-se que os multiplicadores são mapeados para *slices* especiais para processamento DSP (DSP48E). Um *slice* DSP48E contém um multiplicador de 25 x 18 bits, um somador e um acumulador.

A arquitetura foi também sintetizada para *standard-cells* utilizando a tecnologia TSMC 0.18 μ m, usando a ferramenta de síntese Leonardo Spectrum, da Mentor. Os resultados desta síntese são mostrados na Tabela 4.5 para a arquitetura completa e seus principais módulos. Nesta síntese, as memórias SRAM foram retiradas dos módulos de gerenciamento dos vizinhos e memória de predição, mantendo somente a parte lógica. Foram realizadas duas sínteses: a primeira, mantendo os multiplicadores de cálculo do modo plano (mapeando-os para células da biblioteca) e a segunda, substituindo os multiplicadores pelo módulo mostrado na Figura 4.10, que utiliza somente multiplexadores, somadores e deslocamentos, para reduzir área do circuito. A frequência atingida pela arquitetura completa foi de 161 MHz, para ambas as sínteses, resultado este superior às soluções em FPGA. Além disto, a arquitetura sem multiplicadores utilizou 3022 *gates* equivalentes (NAND de 2 entradas) a menos do que a solução com os multiplicadores da biblioteca de células, o que representa uma redução de 6% em termos de número de *gates*.

Tabela 4.5: Resultados de Síntese para *Standard-Cells* TSMC 0.18 μ m.

	Contagem de <i>Gates</i> Equivalentes (NAND de 2 entradas)	Bits de Memória
Gerenciamento das Amostras Vizinhas e Máquina de Controle	20626	30720
Preditor de Amostras (com multiplicadores)	10327	-
Preditor de Amostras (sem multiplicadores)	8295	-
Cálculo SAD	8150	-
Decisão	2172	-
Resíduos	717	-
Reconstrução	3053	-
Memória de Predição (somente as FSMs)	589	12288
Arquitetura Completa (com multiplicadores)	59202	44160
Arquitetura Completa (sem multiplicadores)	56180	

4.5 Análise de Desempenho

A arquitetura deste trabalho foi desenvolvida com o objetivo de processar vídeos da resolução 1080p (1920x1088), com sub-amostragem 4:2:0, a 30 quadros por segundo. Para este requisito é necessário que a arquitetura atinja o desempenho de codificar 244800 macroblocos por segundo. Este requisito de desempenho depende de duas características: o número de ciclos utilizado para codificar um macrobloco; a frequência atingida pela arquitetura na síntese.

A Figura 4.17 mostra como se dá processamento intercalado dos tipos Intra 4x4 (I4MB) e Intra 16x16 (I16MB) na arquitetura desenvolvida e quantos ciclos são necessários para processar um macrobloco. O processamento do modo Intra 4x4 leva 15 ciclos para cada bloco 4x4, após uma etapa de 4 ciclos para leitura da memória e cálculo da disponibilidade dos blocos vizinhos. Este número de ciclos considera que o loop de transformadas e quantização diretas e inversas (T/Q/IT/IQ) processa um bloco no modo Intra 4x4 em 4 ciclos, de acordo com o trabalho de Sampaio et al. (SAMPAIO, 2009). Desta forma, um macrobloco é processado para o modo Intra 4x4 em 244 ciclos.

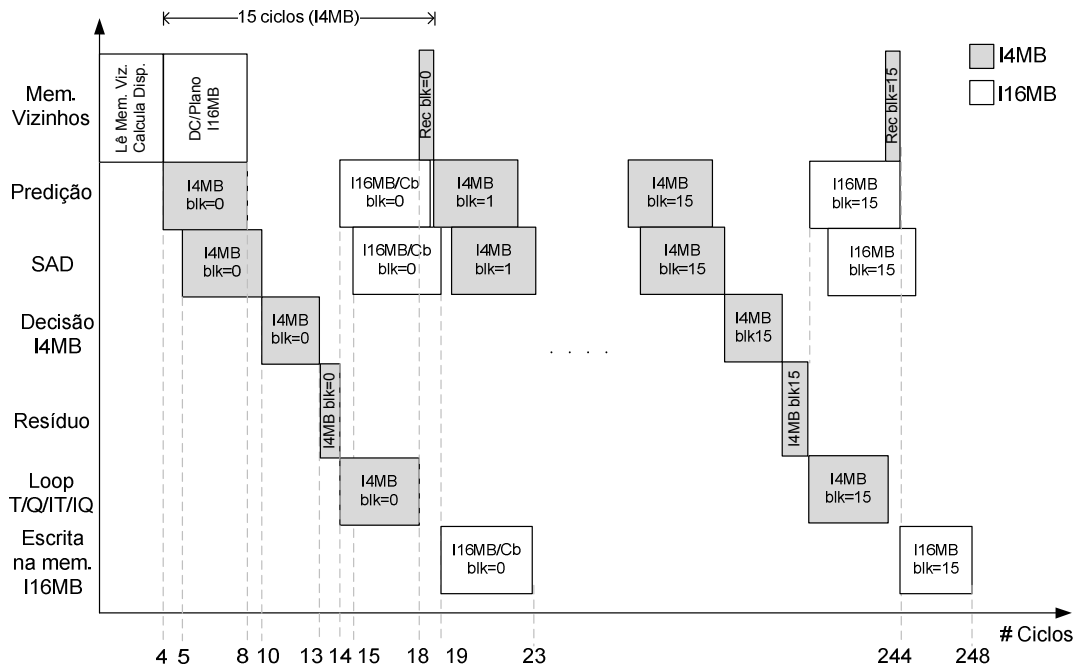


Figura 4.17: Processamento intercalado Intra 4x4 (I4MB) / Intra 16x16 (I16MB).

O processo de predição e cálculo de SAD nos modos Intra 16x16 (I16MB) e crominância (Cb e Cr) são feitos nos intervalos do processamento dos modos Intra 4x4, como mostrado na Figura 4.17. A predição de crominância é feita em paralelo com a predição I16MB, primeiro para Cb e depois para Cr. As amostras resultantes da predição são armazenadas em memória, e os valores de SAD para cada partição do macrobloco são acumulados em registradores. Ao final dos 248 ciclos, o processo de predição e cálculo de SAD para todos os modos é concluído e as amostras dos modos I16MB e crominância são guardadas em memória.

A Figura 4.18 mostra um diagrama do restante do processo de codificação intra-quadro, que envolve as etapas de decisão do modo I16MB e do modo de crominância (o mesmo modo é escolhido para Cb e Cr), decisão I4MB/I16MB e a transformação e quantização dos modos de crominância e do modo I16MB (que não foram implementados neste trabalho), caso este seja escolhido. Pode-se considerar que a transformada/quantização do modo Intra 4x4 escolhido já foi realizada na etapa anterior para todos os blocos e os coeficientes quantizados e podem ser armazenados em memória para posterior codificação, caso o modo Intra 4x4 seja escolhido.

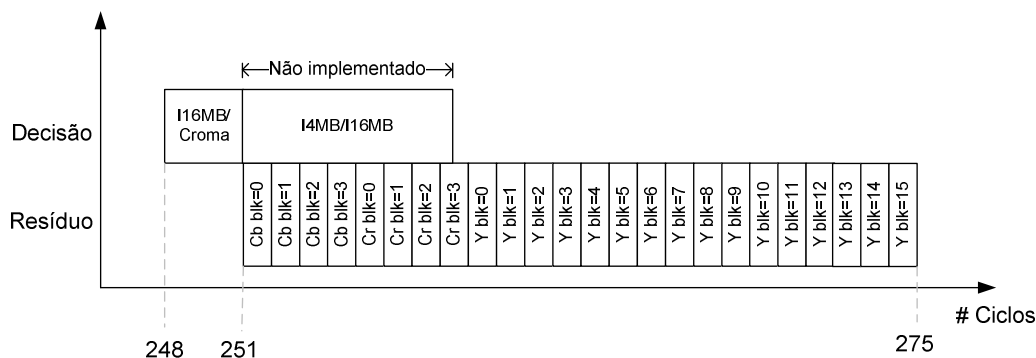


Figura 4.18: Restante do processo de codificação intra-quadro.

A decisão do modo I16MB e do modo de crominância é feita em 3 ciclos. Após isto, a decisão I4MB/I16MB pode ser realizada em paralelo com o cálculo dos resíduos de crominância Cb (e posteriormente Cr) para o modo de crominância escolhido. A decisão I4MB/I16MB não foi tratada neste trabalho. O cálculo dos resíduos pode ser feito em 1 ciclo para cada bloco, pois todas amostras estão armazenadas em memória de uma forma que podem ser lidas com paralelismo de 16 amostras. Considerando o trabalho de Sampaio et al. (SAMPAIO, 2009), 26 ciclos são necessários para processar todos os blocos de crominância (13 para cada canal, Cb e Cr). Mais 29 ciclos são usados para processar os blocos das bordas do modo I16MB, que são os blocos usados para predição dos próximos macroblocos. A transformação/quantização dos blocos de crominância é sempre realizada. Já a transformação/quantização dos blocos no modo I16MB só será feita caso este seja o modo escolhido pela etapa de decisão I4MB/I16MB.

Assim, no melhor caso, a arquitetura leva 275 ciclos ($248 + 27$) para processar um macrobloco caso o modo I4MB seja escolhido. Caso o modo I16MB seja escolhido, a arquitetura leva 304 ciclos ($275 + 29$) para processar as amostras da borda, que serão necessárias para predição do próximo macrobloco.

Considerando o pior caso (304 ciclos), foi calculado o desempenho atingido (em macroblocos/s e quadros/s) pela arquitetura para codificar vídeos em diversas resoluções, operando a uma frequência de 130 MHz, atingida na síntese para dispositivos FPGA Virtex-II Pro. Foi ainda calculado em que frequência a arquitetura precisa operar para codificar os vídeos em diversas resoluções em tempo-real, a 30 quadros/s. Os resultados estão resumidos na Tabela 4.6.

Pelos resultados da tabela, pode-se perceber que a arquitetura desenvolvida ultrapassa os requisitos de desempenho originais, sendo capaz de processar 427631 macroblocos por segundo e até 52 quadros de resolução HD 1080p (1920x1088) por segundo, quando operando a uma frequência de 130 MHz. Além disto, a arquitetura é capaz de processar vídeos na resolução HD 1080p a 30 quadros/s quando operando a uma frequência de 74,4 MHz.

A arquitetura também pode ser usada como parte de um codificador de imagens estáticas (*still image*), processando até 11 quadros de resolução QHDTV (4096x2304) por segundo.

Tabela 4.6: Desempenho da arquitetura para diversas resoluções.

	Requisitos de desempenho		Desempenho atingido pela arquitetura		
	Macroblocos no Quadro	Macroblocos/s (30 quadros/s)	Macroblocos/s (130 MHz)	Quadros/s (130 MHz)	Freq. Mín. 30 quadros/s (MHz)
QHDTV (4096x2304)	36864	983040	427631	11	-
2Kx1K (2048x1024)	8192	245760		52	74,7
HD 1080p (1920x1088)	8160	244800		52	74,4
HD 720p (1280x720)	3600	108000		118	32,8
D1 (720x480)	1350	40500		316	12,3
VGA (640x480)	1200	36000		356	10,9
CIF (352x288)	396	11880		1079	3,6
QCIF (176x144)	99	2970		4319	0,9

4.6 Metodologia de Verificação

A metodologia de verificação utilizada neste trabalho é dividida em duas etapas. Primeiramente, os módulos individuais que compõem a arquitetura são verificados de forma incremental e de acordo com sua funcionalidade, ao decorrer do desenvolvimento. Na segunda etapa, estes módulos são integrados e a arquitetura completa é verificada pela comparação de seus estímulos de saída com a saída de um modelo de referência da predição intra-quadro, tendo como entrada a mesma sequência de vídeo. Este modelo de referência deve ser compatível com a norma do padrão H.264/AVC (ITU-T, 2005).

O JM (SUHRING, 2009) é o *software* que serve como modelo de referência do padrão H.264/AVC, por ter sido desenvolvido em conjunto com a normatização do padrão. No entanto, a forma de implementação não-modular e não-orientada a *hardware* deste *software* torna complexa e demorada a verificação de sub-etapas do processo de codificação, como é o caso da predição intra-quadro.

Assim, um *software* para predição intra-quadro foi descrito, em linguagem MATLAB (MATHWORKS, 2009), para servir como modelo de referência para verificação da arquitetura. O MATLAB foi escolhido por incluir uma série de bibliotecas para leitura e manipulação de vídeos, de forma a tornar mais simples o desenvolvimento do *software*. Este *software* foi verificado comparando as saídas com o JM, para um mesmo vídeo de entrada e usando uma configuração no JM que reflete o que foi implementado no *hardware*. A verificação é feita pela comparação direta da saída da imagem reconstruída do *software* com a saída da imagem reconstruída pelo JM para um mesmo vídeo de entrada.

A Figura 4.19 mostra o resultado gerado pelo *software* de predição intra-quadro, somente para o canal de luminância. A Figura 4.19a mostra um quadro da sequência de vídeo 'Foreman' na resolução QCIF (176x144). As Figuras 4.19b e 4.19c mostram somente os quadros preditos nos modos I4MB e I16MB. Em cada quadro, foram escolhidos somente modos de cada tipo de macrobloco do tipo Intra (I4MB ou I16MB)

e a escolha dentre os modos foi feita pelo menor valor de SAD, usando o mesmo algoritmo implementado pela arquitetura.



Figura 4.19: Resultados do *software* de predição intra-quadro para um quadro da sequência de vídeo ‘Foreman’ (QCIF).

Para verificação, as amostras do quadro original são colocadas em um arquivo texto e utilizadas como entrada do *testbench* para verificação da arquitetura em VHDL, utilizando a ferramenta de simulação ModelSim, da Mentor Graphics. As amostras de saída do quadro predito são colocadas também em arquivo texto e comparadas bit a bit com as amostras do *software* desenvolvido em MATLAB. Foram feitas duas simulações, uma contendo somente macroblocos preditos em I4MB e outra contendo somente macroblocos preditos em I16MB. Este processo de verificação foi realizado de forma que a arquitetura gere um quadro de saída (predito) exatamente igual ao quadro gerado pelo *software*.

Depois de realizada a integração com os módulos de transformadas e quantização diretas e inversas, outro processo de verificação foi feito, desta vez utilizando o JM. Somente o modo Intra 4x4 (I4MB) foi verificado. Funções auxiliares foram implementadas internamente ao JM para extração de estímulos de entrada e saída de cada um dos módulos: predição, SAD, decisão, DCT, quantização direta, quantização inversa, IDCT, reconstrução. O mesmo quadro de entrada ‘Foreman’, utilizado na etapa anterior, foi utilizado como entrada da arquitetura e do JM. O JM foi configurado para utilizar decisão baseada em SAD, para comparação direta com o resultado gerado pela arquitetura.

Depois de corrigidos os erros dos módulos, foram geradas imagens de saída da arquitetura para este quadro de entrada, variando o parâmetro de quantização para ilustrar o funcionamento da predição intra-quadro em conjunto com os módulos de transformadas/quantização. As imagens geradas estão ilustradas na Figura 4.20.

A Figura 4.20a é o quadro predito usando uma codificação sem perdas (QP = 0) para ilustrar somente a qualidade da predição. Na Figura 4.20b é mostrado o quadro reconstruído, utilizando esta mesma predição sem perdas. Percebe-se que, após a reconstrução, o PSNR é consideravelmente maior. A Figura 4.20c mostra um resultado real, com perdas, utilizando QP = 28 (padrão do arquivo de configuração do JM). A verificação para os modos I16MB e croma não foi realizada em conjunto com os módulos de transformadas/quantização, sendo deixada como trabalho futuro.



a) Quadro Predito (I4MB).
QP = 0; PSNR = 27,89 dB.

b) Quadro Reconstituído (I4MB).
QP = 0; PSNR = 65,88 dB.

c) Quadro Reconstituído (I4MB).
QP = 28; PSNR = 37 dB.

Figura 4.20: Resultados de saída das arquiteturas de predição intra-quadro, transformadas e quantização diretas e inversas (somente I4MB) para um quadro da sequência de vídeo 'Foreman' (QCIF) em diferentes parâmetros de quantização (QP).

5 COMPARAÇÕES COM TRABALHOS RELACIONADOS

5.1 Revisão de Trabalhos da Literatura

A arquitetura desenvolvida neste trabalho foi comparada com diversos trabalhos similares encontrados na literatura. A grande maioria dos trabalhos apresentam codificadores intra-quadro compatíveis com o padrão H.264/AVC, contendo a predição intra-quadro, foco deste trabalho, e os módulos de transformadas e quantização (diretas e inversas) e codificação de entropia do tipo CAVLC. Somente o trabalho de Kuo e Lin (KUO, 2008) apresenta uma arquitetura para o codificador intra-quadro com codificação de entropia CABAC.

Como esta dissertação somente tratou do módulo de predição intra-quadro para codificadores do padrão H.264/AVC, algumas comparações foram feitas especificamente com a parte de predição intra-quadro dos trabalhos relacionados. Nas seções a seguir está descrito um breve resumo de cada trabalho a ser comparado e, posteriormente, as tabelas de comparações são mostradas.

5.1.1 Trabalho de Huang et al.

O trabalho de Huang et al. (HUANG, 2004; HUANG, 2005) foi um dos primeiros a apresentar uma solução em *hardware* para a codificação intra-quadro compatível com o padrão H.264/AVC, envolvendo os módulos de predição intra-quadro, transformadas e quantização diretas e inversas e codificador de entropia CAVLC.

Primeiramente, o trabalho faz uma análise do impacto do algoritmo de codificação implementado em *software* executando em uma arquitetura centrada em processador do tipo RISC, comparando com uma arquitetura de *hardware* que explore o paralelismo inerente do algoritmo. O ganho em desempenho da arquitetura de *hardware* específica pode chegar a 215 vezes em termos de ciclos de relógio.

Com base nisto, o trabalho propõe uma arquitetura de *hardware* para o codificador intra-quadro com 2 macro-estágios de processamento de macrobloco, separando a etapa de codificação de entropia das demais etapas que envolvem o laço de codificação intra-quadro (predição, transformadas e quantização diretas e inversas). Devido à dependência de dados no processamento da predição Intra 4x4 (I4MB), este trabalho propõe o escalonamento no processamento I4MB/I16MB que serviu de base para a arquitetura desenvolvida no contexto desta dissertação. O trabalho de Huang et al.

também propõe um preditor de amostras reconfigurável, ou seja, gera a predição para 4 amostras dependendo do modo de predição, seja ele do tipo Intra 4x4, Intra 16x16 ou crominância. A decisão de modo é baseada na saída da DCT, ou seja, cada modo de predição é processado pela DCT, semelhante ao que é feito no *software* de referência do padrão H.264/AVC (JM). Este algoritmo não sofre nenhuma perda de qualidade em comparação com o JM (HUANG, 2005).

A arquitetura foi sintetizada para a tecnologia TSMC 0.25 μ m 1P5M e atingiu a máxima frequência de 55 MHz. Como a arquitetura leva no máximo 1300 ciclos para codificar um macrobloco, ela atinge o desempenho para codificar 16 Mega-pixels em 1 segundo, ou 31 quadros de resolução D1 (720x480) por segundo. Demais características da arquitetura são apresentadas nas Tabelas 5.1 e 5.2. Um trabalho posterior deste mesmo autor (HUANG, 2005a) apresenta a arquitetura de um codificador H.264/AVC completo, implementado em TSMC 0.18 μ m, que atinge desempenho para codificar vídeos HDTV 720p a 30 quadros por segundo.

5.1.2 Trabalho de Suh, Park e Cho

O trabalho de Suh, Park e Cho (SUH, 2005) propõe uma nova arquitetura para a codificação intra-quadro, sendo que a principal contribuição é reduzir o número de ciclos para o processamento do modo Intra 16x16 (I16MB) em relação ao trabalho de Huang (HUANG, 2005). A técnica usada é associar uma transformada Hadamard ao processo de decisão do modo de predição e reusar este cálculo para os coeficientes DC (que já são processados pela Hadamard caso a predição seja Intra 16x16), guardando estes valores em *buffer* ao invés de reprocessá-los, economizando 178 ciclos.

A arquitetura leva 927 ciclos para processar um macrobloco, sendo 549 ciclos para o modo Intra 4x4, e os 378 ciclos restantes para processar os modos Intra 16x16, Cb e Cr. Este trabalho otimiza o número de ciclos/macrobloco ao custo de um aumento na área do circuito, pela duplicação dos caminho de dados, um para Intra 4x4 e outro para Intra 16x16/Croma. O hardware proposto utiliza 192 Kgates na tecnologia TSMC 0.25 μ m.

5.1.3 Trabalho de Cheng, Ku e Chang

O trabalho de Cheng, Ku e Chang (CHENG, 2006) apresenta três contribuições, principalmente para redução de área do codificador intra-quadro em *hardware*. A primeira contribuição proposta é de retirar o modo plano do codificador, pois, segundo algumas simulações com sequências de vídeo reais, o modo plano é pouco escolhido e representa pouco ganho de qualidade. A segunda proposta é um novo algoritmo de cálculo do custo e decisão, baseado na transformada 4x4 inteira associada a uma matriz de coeficientes de escala pré-determinados. Este cálculo do custo compensa de certa forma a perda de qualidade ocasionada pela retirada do cálculo do modo plano. A terceira proposta é um conjunto de técnicas arquiteturais para minimizar as “bolhas” no processo de codificação, como a reutilização do hardware de predição para todos os modos de predição e reprocessamento dos modos Intra 16x16/croma para redução de *buffers*, entre outras. Só é usada memória intra-chip para guardar os coeficientes quantizados (13312 bits) e as amostras do macrobloco original (3072 bits).

Com isto, a arquitetura utilizou somente 92 Kgates na tecnologia UMC 0.18 μ m, sendo capaz de processar vídeos HD 720p a 30 quadros/s na frequência de 117 MHz, pois processa um macrobloco em menos de 1080 ciclos.

5.1.4 Trabalho de Ku et al.

O trabalho de Ku et al. (KU, 2006) é uma continuação do trabalho anterior de Cheng, Ku e Chang (CHENG, 2006), e propõe um codec intra-quadro com CAVLC. Os resultados da parte do codificador são similares, mas a área de *hardware* da arquitetura completa é maior devido à inserção da parte de decodificação.

5.1.5 Trabalho de Li et al.

O trabalho de Li et al. (LI, 2007) também é uma continuação do trabalho de Cheng (CHENG, 2006) e propõe um novo algoritmo de três passos para decisão de modos Intra 4x4 e uma arquitetura com paralelismo variável, ou seja, paralelismo de 8 amostras para predição/DCT/DHT, e de 4 amostras para a etapa de quantização. A arquitetura é composta por duas unidades de predição intra-quadro de forma a aumentar o *throughput* e reduzir o número de ciclos desta etapa crítica.

A arquitetura foi otimizada para redução de área e consumo de potência, pela diminuição dos estágios de pipeline e aumento do caminho crítico. Desta forma, ela atingiu somente 62,5 MHz implementada na tecnologia UMC 0.18 μ m, mas necessita de 61 MHz para processar vídeos HD 720p a 30 quadros/s. Não ficou claro neste artigo se o modo plano foi implementado, pois ele tinha sido excluído em (CHENG, 2006).

5.1.6 Trabalho de Chang et al.

A principal contribuição do trabalho de Chang et al. (CHANG, 2007) é uma arquitetura de codificação intra-quadro com suporte para diferentes níveis de qualidade da imagem na saída, configurando dinamicamente a frequência de operação a partir do consumo de potência desejado para cada tipo de aplicação.

São definidos três níveis de qualidade (QS0, QS1 e QS2, sendo QS0 a melhor qualidade) que variam o número de ciclos da etapa de decisão do modo. A arquitetura é capaz de processar vídeos HD 720p a 30 quadros/s nas frequências 70 e 85 MHz e D1 a 26, 33 e 47 MHz, levando no mínimo 626 ciclos (QS2) e no máximo 1112 ciclos (QS0) para processar um macrobloco. A configuração de qualidade QS2 tem uma perda de qualidade de 0.057 dB (PSNR) em relação ao QS0 (mesmo decisão do JM), o que resulta em 6,63% de acréscimo de bits para o mesmo PSNR.

A arquitetura foi implementada em tecnologia TSMC 0.13 μ m e ocupou 169 Kgates, sendo que a predição intra-quadro incluindo transformadas e quantização ocupam cerca de 80 Kgates.

5.1.7 Trabalho de Sahin e Hamzaoglu

O trabalho de Sahin e Hamzaoglu (SAHIN, 2007) é um dos poucos trabalhos encontrados que apresenta resultados para FPGA. A contribuição deste trabalho é mais voltada a aplicações de baixa resolução, pois apresenta uma arquitetura com um baixo consumo de recursos de hardware, voltada para o processamento em tempo real de vídeos VGA (640x480), a 27 quadros/s. A solução usada para atingir o mínimo consumo é explorar a serialidade das operações, ou seja, a arquitetura leva no máximo 2640 ciclos para processar um macrobloco.

Desta forma, a arquitetura só atinge desempenho para processar 27 quadros VGA por segundo, operando a uma frequência de 90 MHz no dispositivo XC2V8000. A arquitetura ocupou somente 1001 *slices*, ou 2002 *function generators* (equivalente as

LUTs), o que representa apenas 2% das células lógicas (CLB) do dispositivo XC2V8000.

5.1.8 Trabalho de Kuo e Lin

O trabalho de Kuo e Lin (KUO, 2008) foi o primeiro a propor uma arquitetura para o codificador intra-quadro com capacidade de processar vídeos com resolução HD 1080p em tempo-real, a 30 quadros/s. Diferentemente de (LI, 2007) e (CHANG, 2007), o trabalho propõe uma implementação completa do codificador intra-quadro, com todos os modos de predição intra suportados pelo padrão, decisão baseada em SATD (*Sum of Absolute Transformed Differences*) e codificador de entropia CABAC.

O trabalho propõe uma nova ordem para predição dos blocos no modo Intra 4x4 e utiliza dois módulos de predição para processar dois blocos 4x4 em paralelo. A unidade de cálculo de custo também é duplicada. Esta otimização resulta na utilização de 562 ciclos, no total, para codificar um macrobloco. A arquitetura foi implementada em tecnologia TSMC 0.13 μ m e ocupou 212 Kgates no total (199K para o core e 12K para a interface de barramento). O número de *gates* para comparação direta com este trabalho não foi informado. A arquitetura atinge 142 MHz de frequência, mas é capaz de processar vídeos HD 1080p a 30 quadros/s operando em 138 MHz.

5.1.9 Trabalho de Lin et al.

O trabalho de Lin et al. (LIN, 2009) é uma continuação dos trabalhos de (LI, 2007), (CHENG, 2006) e (KU, 2006), com poucas modificações arquiteturais para o suporte a vídeos HD 1080p a 30 quadros/s. Ela implementa o mesmo algoritmo de três passos para decisão dos modos 4x4, o que resulta em somente 20 ciclos para decisão de cada bloco 4x4. Neste trabalho ficou clara a remoção do modo plano e a decisão baseada em um método aprimorado SATD, proposto em (CHENG, 2006). A degradação média é de 0.11-dB em termos de PSNR.

A arquitetura foi implementada em TSMC 0.13 μ m e utilizou 94,7 Kgates, atingindo uma frequência de 140 MHz, sendo capaz de codificar vídeos de resolução HD 1080p a 30 quadros/s nesta frequência de operação. Além disto, a arquitetura é capaz de codificar vídeos QHDTV (4096x2304) a 6,78 quadros/s. Isto é útil para aplicações de codificação de imagens estáticas, usado em câmeras digitais.

5.2 Comparações com a Arquitetura Desenvolvida

A arquitetura proposta e desenvolvida neste trabalho foi comparada com os trabalhos relacionados em termos de uso de recursos de *hardware* (contagem de *gates*, para resultados em *standard-cells* e LUTs utilizadas, para resultados em FPGA) e em termos de desempenho atingido para codificar quadros do tipo Intra em tempo-real, a 30 quadros por segundo. A comparação do uso de recursos de *hardware* está mostrada na Tabela 5.1. A Tabela 5.2 compara os resultados de desempenho atingido com os trabalhos relacionados.

Tabela 5.1: Resultados comparativos em termos de uso de recursos de *hardware*.

	Tecnologia	Área Total (<i>Gates</i> ou LUTs)	Área Intra (<i>Gates</i> ou LUTs) ¹	Memória Intra-Chip (Bits) ¹	Modos Intra Suportados
(HUANG, 2005)	TSMC 0.25 μ m	84 K	43 K	5120	Todos
(SUH, 2005)	Hynix 0.35 μ m	192 K	114 K	8912	Todos
(CHENG, 2006)	UMC 0.18 μ m	92 K	38 K	-	Exceto Plano
(KU, 2006)	UMC 0.18 μ m	103 K	36 K	-	Exceto Plano
(LI, 2007)	UMC 0.18 μ m	72 K	N/A	-	Não informa
(CHANG, 2007)	TSMC 0.13 μ m	169 K	80 K ³	5120	Todos
(SAHIN, 2007)	FPGA Virtex-2 ²	2002 LUTs	N/A	-	Todos
(KUO, 2008)	TSMC 0.13 μ m	199 K	N/A	-	Todos
(LIN, 2009)	TSMC 0.13 μ m	94,7 K	43 K	-	Exceto Plano
Arquitetura Desenvolvida	TSMC 0.18 μ m	56 K		44160	Todos
	FPGA Virtex-2P	8866 LUTs			
	FPGA Virtex-5	7516 LUT/Flop pairs			

¹ Resultados calculados com base em dados fornecidos, para comparação direta com este trabalho

² Dispositivo Xilinx Virtex-2 XC2V8000-5

³ Inclui transformadas/quantização diretas e inversas

Tabela 5.2: Resultados comparativos em termos de desempenho.

	Frequência Máxima (MHz)	Ciclos/MB	Menor Frequência para Processamento em Tempo-Real a 30 Quadros/s (MHz)			
			1080p	720p	D1	CIF
(HUANG, 2005)	55	< 1300	-	-	54	16
(SUH, 2005)	54	927	-	-	38	11
(CHENG, 2006)	125	< 1080	-	117	43	12
(KU, 2006)	125	< 1080	-	117	43	12,8
(LI, 2007)	62,5	< 560	-	61	23	6,7
(CHANG, 2007)	130	1112 / 626	-	70 / 85	26 / 47	N/A
(SAHIN, 2007)	90	< 2640	-	-	-	33
(KUO, 2008)	142	< 562	138	61	23	6,7
(LIN, 2009)	140	560	140	61	23	6,7
Arquitetura Desenvolvida	161 (0.18 μ m)	< 304	74,4	32,8	12,3	3,6
	130 (Virtex-2)					
	135 (Virtex-5)					

Todos os trabalhos relacionados apresentam dados para o codificador intra-quadro completo em *hardware*, de forma que a comparação do uso de recursos de *hardware* com o presente trabalho, que trata somente da etapa predição (sem transformadas/quantização e codificação de entropia), pode ser tendenciosa. Desta forma, em alguns casos foi possível extrair, a partir de dados contidos nos artigos, somente o número de *gates* para os módulos compatíveis com os que foram implementados neste trabalho, como mostrado nas colunas “Área Intra” e “Memória Intra-Chip” da Tabela 5.1. Nota-se que a arquitetura desenvolvida consome mais recursos de *hardware* que as demais, sendo menor somente que a proposta em (SUH, 2005). Isto acontece devido à decisão de atingir alta vazão de dados pela exploração do paralelismo de *hardware*. As propostas de (CHENG, 2006) e (KU, 2006) são as que consomem menos recursos, mas não implementam o modo plano, que é o mais complexo dos modos de predição intra. Isto pode levar a uma perda de qualidade da

imagem para certos casos, em comparação com soluções que suportam todos os modos Intra. Em comparação com a solução de (LIN, 2009), que tem desempenho para HD 1080p em tempo-real, a arquitetura tem um acréscimo de 30% em termos de contagem de *gates*.

Além disto, a arquitetura desenvolvida consome mais bits de memória que as demais arquiteturas. A maior parcela de memória utilizada está relacionada com a memória de amostras vizinhas (30720 bits), que armazena uma linha de um quadro de resolução HD 1080p. O trabalho de (HUANG, 2005) utiliza esta mesma técnica, mas usa menos memória já que não suporta resolução HD 1080p ou HD 720p, somente D1. Existe ainda a memória de predição (12288 bits), que foi utilizada para economizar os ciclos de recálculo do modo I16MB.

Relativo aos resultados de desempenho, mostrados na Tabela 5.2, a arquitetura atinge a maior frequência de operação e o menor número de ciclos para processar um macrobloco (Ciclos/MB), atingindo uma alta vazão de dados (*throughput*) quando comparado com os trabalhos relacionados. A arquitetura foi sintetizada para TSMC 0.18 μ m, tecnologia compatível com a maioria dos trabalhos relacionados que apresentam resultados em *standard-cells*, com exceção de (HUANG, 2005) e (SUH, 2005) que apresentam resultados em tecnologias mais antigas e, portanto, resultados de frequência de operação não podem ser comparados diretamente. Somente os trabalhos de (KUO, 2008) e (LIN, 2009) e este trabalho são capazes de codificar vídeos HD 1080p em tempo real (contendo somente quadros do tipo Intra). A arquitetura conseguiu diminuir o número de ciclos para processar um macrobloco em 45% em relação às melhores soluções encontradas, de (LI, 2007) e (LIN, 2009), com um acréscimo de 30% em número de *gates* em comparação com (LIN, 2009). Este ganho é atingido devido ao elevado grau de paralelismo da etapa de predição, ao intercalamento da predição I4MB/I16MB e ao uso de memória de predição para ganhar ciclos em relação ao recálculo das amostras preditas.

Devido à elevada vazão de dados, a arquitetura desenvolvida pode processar quadros em tempo-real (30 quadros por segundo) a uma frequência menor que todos os trabalhos relacionados, uma redução de 46% em comparação com a frequência necessária em (KUO, 2008) para codificação em tempo real, melhor trabalho encontrado na literatura neste sentido. Além disto, a arquitetura pode processar até 11 quadros de resolução QHDTV (4096x2304) por segundo operando a uma frequência de 130 MHz, enquanto que o trabalho de (LIN, 2009) somente processa 6,78 quadros/s operando a uma frequência superior (140 MHz).

A arquitetura aqui apresentada foi publicada no IEEE International Conference on Multimedia and Expo (DINIZ, 2009).

6 CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho apresentou inicialmente um breve estudo sobre o padrão H.264/AVC de compressão de vídeo e sobre algoritmos para codificação intra-quadro e focou a proposta e desenvolvimento de uma arquitetura de predição intra-quadro em *hardware* para ser integrada a um codificador compatível com o padrão H.264/AVC no perfil *Main*. O requisito inicial de desempenho, de codificar vídeos de alta definição em tempo-real (mais especificamente vídeos HD 1080p a 30 quadros/s), foi atingido. Desta forma, os objetivos definidos inicialmente no trabalho foram alcançados.

Foi realizada uma investigação de soluções para guiar o desenvolvimento da arquitetura de predição intra-quadro em *hardware*. Das duas soluções arquiteturais investigadas, a segunda solução arquitetural foi escolhida. Esta solução utiliza um esquema de tempo compartilhado para o caminho de predição intra-quadro, baseado no trabalho de Huang et al. (HUANG, 2005), o que resolve o problema de latência causado pela dependência de dados da predição intra-quadro com as etapas de transformadas e quantização diretas e inversas.

Posteriormente, esta arquitetura foi desenvolvida em VHDL e sintetizada para dispositivos FPGAs da Xilinx, das famílias Virtex-II Pro e Virtex-5, e para *standard-cells*, na tecnologia TSMC 0.18 μ m. Resultados de síntese mostraram que a arquitetura de predição intra-quadro atinge uma frequência de 130 MHz e ocupa 8866 das LUTs e 11 BRAMs do dispositivo XC2VP30 da família Virtex-II Pro da Xilinx. Para o dispositivo XC5VLX110, da família Virtex-5 da Xilinx, a arquitetura atinge 135 MHz e ocupa 7516 LUTs e 11 BRAMs. A arquitetura foi também sintetizada para *standard-cells* utilizando a tecnologia TSMC 0.18 μ m. Duas sínteses foram realizadas: a primeira mantendo os multiplicadores de cálculo do modo plano (mapeando-os para células da biblioteca), e a segunda, substituindo os multiplicadores pelo módulo que utiliza somente multiplexadores, somadores e deslocamentos, para reduzir área do circuito. A arquitetura atingiu uma frequência de 161 MHz. A arquitetura utiliza 59K e 56K *gates* equivalentes (NAND de 2 entradas), para a primeira e segunda síntese, respectivamente.

Uma metodologia de verificação funcional foi aplicada, utilizando um quadro de uma sequência de vídeo real, e usou como base um *software* descrito em MATLAB, desenvolvido no escopo deste trabalho, e o software de referência do padrão H.264/AVC, o JM (SUHRING, 2009). Os resultados de imagem foram gerados e a qualidade visual foi medida em termos de PSNR, mostrando que a arquitetura está correta considerando o subconjunto de estímulos de entrada testados.

Uma análise de desempenho mostrou, com base em trabalhos paralelos do grupo de pesquisa (SAMPAIO, 2009), que a arquitetura atinge o desempenho para codificar vídeos de resolução HD 1080p, utilizando somente quadros do tipo Intra, a uma taxa de 52 quadros por segundo, superior aos 30 quadros por segundo requeridos. A arquitetura usa somente 304 ciclos no pior caso. Isto foi possível devido à exploração do paralelismo para cálculo dos modos de predição, aliado ao esquema de tempo compartilhado na etapa de predição que resolve o problema da latência na codificação intra-quadro. O custo da exploração do paralelismo é o aumento no número de recursos de *hardware* usados em comparação com os trabalhos relacionados. Porém, o uso da solução de tempo compartilhado consegue reduzir o hardware em comparação com a solução de Suh, Park e Cho (SUH, 2005). Comparando com a solução de Lin et al. (LIN, 2009), que atinge desempenho para codificar HD 1080p a 30 quadros/s, a arquitetura proposta tem um acréscimo de 30% no número de *gates* equivalentes na implementação ASIC.

Apesar do aumento no uso de recursos de *hardware*, a arquitetura desenvolvida no presente trabalho supera, em termos de desempenho, todos os trabalhos da literatura aqui listados. Os resultados mostraram que a arquitetura conseguiu reduzir o número de ciclos para processar um macrobloco Intra em 45% quando comparado aos melhores trabalhos encontrados neste aspecto, o de Li et al. (LI, 2007) e Lin et al. (LIN, 2009). Aliado a isto, a arquitetura pode atingir o objetivo inicial de processar vídeos de resolução HD 1080p a 30 quadros/s a 74,4 MHz, frequência esta muito menor que a máxima frequência de síntese. Esta frequência é 46% menor quando comparada ao trabalho de Kuo e Lin (KUO, 2008), melhor trabalho encontrado neste sentido. Utilizar uma frequência de operação menor é vantajoso no aspecto de diminuir o consumo de potência do *hardware* e, por consequência, de energia. Além disto, a arquitetura pode ser usada em outras aplicações, como codificação de imagens estáticas, podendo processar resoluções altas como QHDTV (4096x2304) com taxa de até 11 quadros/s.

Em vista destes resultados, a arquitetura torna-se útil para ser integrada em um codificador compatível H.264/AVC perfil *Main*, com desempenho para codificar vídeos de alta definição em tempo-real. Esta tarefa é deixada como trabalho futuro. Outros trabalhos futuros são listados na próxima seção.

6.1 Trabalhos Futuros

- As etapas de geração do modo DC I16MB e parâmetros do modo plano (no módulo de gerenciamento das amostras vizinhas e máquina de controle) consumiram mais recursos do FPGA do que o previsto. Foi utilizada a ferramenta de síntese Xilinx ISE 8.2i para realizar a síntese para o dispositivo XC2VP30, da família Virtex-II Pro. Alguns experimentos foram conduzidos no sentido de substituir algumas diretivas *variables* por *signals*, alterando no código o necessário para que a função do módulo fosse mantida. Verificou-se que o uso de recursos do FPGA diminuiu, ou seja, a melhor hipótese foi que a ferramenta de síntese não conseguiu reutilizar os operadores em estados diferentes da máquina de estados. Pretende-se, como trabalho futuro, implementar toda máquina de estados utilizando somente diretivas *signals*, avaliando os resultados em termos de uso de recursos do FPGA. Outra estratégia é realizar este experimento em outras ferramentas de síntese para FPGA e ASIC;

- Avaliar o impacto da arquitetura caso não seja utilizada a memória de predição, realizando o recálculo das amostras preditas para codificação;
- Investigar estratégias de redução de área em conjunto com módulos de transformadas/quantização;
- Verificação da arquitetura para os modos Intra 16x16 e croma; e crominância;
- Comparar a qualidade do vídeo gerado com os trabalhos relacionados para diversos parâmetros de quantização;
- Integrar a arquitetura em um codificador completo compatível com o H.264/AVC perfil *Main*;
- Estender a arquitetura de predição intra-quadro para suportar outros perfis ou extensões do padrão, como os perfis High e SVC.

REFERÊNCIAS

AGOSTINI, L. V., **Desenvolvimento de Arquiteturas de Alta Performance Dedicadas à Compressão de Vídeo Segundo o Padrão H.264**. Tese (Doutorado) - Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, RS, 2007.

AGOSTINI, L. V.; AZEVEDO, A.; STAEHLER, W.; ROSA, V.; ZATT, B.; PINTO, A. C.; PORTO, R. E. C.; BAMPI, S.; SUSIN, A. Design and FPGA Prototyping of a H.264/AVC Main Profile Decoder for HDTV. **Journal of the Brazilian Computer Society**, [S.l.], v. 12, n.1, p. 25-36, 2007a.

AZEVEDO, A. P. **MoCHA: Arquitetura Dedicada para a Compensação de Movimento em Decodificadores de Vídeo de Alta Definição, Seguindo o Padrão H.264**. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre, RS, 2006.

CHANG, C.-H.; et al. A Quality Scalable H.264/AVC Baseline Intra Encoder for High Definition Video Applications. In: Workshop on Signal Processing Systems (SiPS), IEEE, 2007. **Proceedings...** Shanghai: IEEE, 2007.

CHENG, C.-C.; KU, C.-W.; CHANG, T.-S. A 1280x720 Pixels 30 Frames/s H.264/MPEG-4 AVC Intra Encoder. In: International Symposium on Circuits and Systems (ISCAS), IEEE, 2006. **Proceedings...** Island of Kos: IEEE, 2006.

DINIZ, C. M.; ZATT, B.; AGOSTINI, L. V.; SUSIN, A. A.; BAMPI, S. A Real Time H.264/AVC Intra Frame Prediction Hardware Architecture for HDTV 1080p Video. In: International Conference on Multimedia and Expo (ICME), IEEE, 2009. **Proceedings...** New York City: IEEE, 2009.

GHANBARI, M. **Standard Codecs: Image Compression to Advanced Video Coding**. United Kingdom: The Institute of electrical Engineers, 2003.

HUANG, Y.-W.; et al. Hardware Architecture Design for H.264/AVC Intra Frame Coder. In: International Symposium on Circuits and Systems (ISCAS), IEEE, 2004. **Proceedings...** Vancouver: IEEE, 2004.

HUANG, Y.-W.; et al. Analysis, Fast Algorithm, and VLSI Architecture Design for H.264/AVC Intra Frame Coder. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.15, n.3, p. 378-401, mar. 2005.

HUANG, Y.-W.; et al. A 1.3TOPS H.264/AVC Single-Chip Encoder for HDTV Applications. In: International Solid-State Circuits Conference (ISSCC), IEEE, 2005. **Proceedings...** San Francisco: IEEE, 2005a.

ISO – International Organization for Standardization. Disponível em: <<http://www.iso.org>>. Acesso em: novembro 2008.

ISO – International Organization for Standardization. MPEG – Moving Pictures Working Group. Disponível em: <<http://www.chiariglione.org/mpeg>>. Acesso em: novembro 2008a.

ISO – International Organization for Standardization. ISO/IEC 11172-2:1993 Information technology -- Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s -- Part 2: Video. [S.l.], 1993.

ITU – INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Home**. Disponível em: <www.itu.int/ITU-T/>. Acesso em: novembro 2008.

ITU – INTERNATIONAL Telecommunication Union. Joint Video Team (JVT). Disponível em: <<http://www.itu.int/ITU-T/studygroups/com16/jvt/>>. Acesso em: novembro 2008a.

ITU – INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation H.264/AVC (11/07)**: advanced video coding for generic audiovisual services. [S.l.], 2007.

ITU – INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation H.264/AVC (03/05)**: advanced video coding for generic audiovisual services. [S.l.], 2005.

ITU – INTERNATIONAL TELECOMMUNICATION UNION. **JVT-L050**: H.264/AVC AVC Fidelity Range Extensions. [S.l.], 2004.

ITU – INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation H.264/AVC (05/03)**: advanced video coding for generic audiovisual services. [S.l.], 2003.

ITU – INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation H.263 v3 (11/00)**: video coding for low bit rate communication. [S.l.], 2000.

ITU – INTERNATIONAL TELECOMMUNICATION UNION. **Information technology - Digital compression and coding of continuous-tone still images - Requirements and guidelines (T.81)**. [S.l.], 1992.

KU, C.-W.; et al. A High-Definition H.264/AVC Intra-Frame Codec IP for Digital Video and Still Camera Applications. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.16, n.8, p. 917-928, ago. 2006.

KUO, H.-C.; LIN, Y.-L. An H.264/AVC Full-Mode Intra-Frame Encoder for 1080HD Video. In: International Conference on Multimedia and Expo (ICME), IEEE, 2008. **Proceedings...** Hannover: IEEE, 2008.

- LI, D.-W.; et al. A 61MHz 72K Gates 1280x720 30FPS H.264 Intra Encoder. In: International Conference on Acoustic, Speech and Signal Processing (ICASSP), IEEE, 2007. **Proceedings...** Honolulu : IEEE, 2007.
- LIN, C.-C.; et al. A 160K Gates/4.5 KB SRAM H.264 Video Decoder for HDTV Applications. **IEEE Journal of Solid-State Circuits**, [S.l.], v.42, n.1, p. 170-182, jan. 2007.
- LIN, Y.-K.; et al. A 140-MHz 94 K Gates HD 1080p 30-Frames/s Intra-Only Profile H.264 Encoder. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.19, n.3, p. 432-436, mar. 2009.
- MATHWORKS INC. MATLAB and Simulink for Technical Computing. Disponível Disponível em: < www.mathworks.com>. Acesso em: mai. 2009.
- PAN, F.; et al. Fast Intra Mode Decision Algorithm For H.264/AVC Video Coding. In: International Conference on Image Processing (ICIP), IEEE, 2004. **Proceedings...**Singapore: IEEE, 2004.
- PORTO, R. **Desenvolvimento Arquitetural para Estimação de Movimento de Blocos de Tamanhos Variáveis Segundo o Padrão H.264/AVC de Compressão de Vídeo Digital**, 2008. 96f. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- PURI, A. et al. **Video Coding Using the H.264/MPEG-4 AVC Compression Standard**. Elsevier Signal Processing: Image Communication, [S.l.], n. 19, p.793–849, 2004.
- RICHARDSON, I. **Video Codec Design – Developing Image and Video Compression Systems**. Chichester: John Wiley and Sons, 2002.
- RICHARDSON, I. **H.264/AVC and MPEG-4 Video Compression – Video Coding for Next-Generation Multimedia**. Chichester: John Wiley and Sons, 2003.
- ROSA, V. S. **Projeto de Arquiteturas de Hardware para a Compressão de Vídeo no Padrão H.264/AVC**. Proposta de Tese (Doutorado) - Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, RS, 2009.
- SAMPAIO, F.; PALOMINO, D.; DORNELLES, R.; AGOSTINI, L.; BAMPI, S. Arquitetura Dedicada para o Loop de Transformadas e Quantização para a Predição Intra-Quadros do Padrão H.264/AVC. In: XV Workshop Iberchip (IWS), 2009. **Proceedings...** Buenos Aires, 2009.
- SAHIN, E.; HAMZAOGLU, I.; An Efficient Hardware Architecture for H.264 Intra Prediction Algorithm. In: Design, Automation & Test in Europe Conference and Exhibition (DATE), IEEE, 2007. **Proceedings...**Nice Acropolis: IEEE, 2007.
- STAEHLER, W. T.; BERRIEL, E. A.; SUSIN, A. A.; BAMPI, S. Architecture of an HDTV Intraframe Predictor for a H.264 Decoder. In: IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), IEEE, 2006. **Proceedings...** Nice: IEEE, 2006.

SUHRING, K. H.264/AVC Reference Software. **In: Fraunhofer Heinrich-Hertz-Institute**. Disponível em: <<http://iphome.hhi.de/suehring/tml/download/>>. Acesso em: mai. 2009.

SUH, K.; PARK, S.; CHO, H. An Efficient Hardware Architecture of Intra Prediction and TQ/IQIT Module for H.264 Encoder. **ETRI Journal**, [S.l.], v. 27, n. 5, p. 511-524, out. 2005.

SULLIVAN, G.; WIEGAND, T. Rate-Distortion Optimization for Video Compression. **IEEE Signal Processing Magazine**, [S.l.], v. 15, p. 74-90, Nov. 1998.

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, SPIE, 2004. **Proceedings...** Denver: SPIE, 2004.

SUNNA, P. AVC / H.264/AVC – An Advanced Video Coding System for SD and HD Broadcasting. **European Broadcasting Union Technical Review**, [S.l.], n. 302, Apr. 2005. Disponível em: <http://www.ebu.ch/en/technical/trev/trev_302-sunna.pdf>. Acesso em: set. 2009.

TSAI, A.-C.; PAUL, A.; WANG, J.-C. WANG, J.-F. Intensity Gradient Technique for Efficient Intra-Prediction in H.264/AVC. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.18, n.5, p. 694-698, mai. 2008.

WIEGAND, T. et al. Rate-Constrained Coder Control and Comparison of Video Coding Standards. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v. 13, n. 7, p. 688-703, July 2003.

XILINX INC. Xilinx: The Programmable Logic Company. Disponível em: <www.xilinx.com>. Acesso em: mai. 2009.

XILINX INC. Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet.[S.l.], 2007. Disponível em: <www.xilinx.com>. Acesso em: mai. 2009a.

XILINX INC. Virtex-5 Family Overview.[S.l.], 2009. Disponível em: <www.xilinx.com>. Acesso em: mai. 2009b.

ZATT, B.; DINIZ, C. M.; AGOSTINI, L. V. SUSIN, A. A.; BAMPI, S. SystemC Modeling of an H.264/AVC Intra Frame Encoder Architecture. In: IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), IEEE, 2008. **Proceedings...** Rhodes Island: IEEE, 2008.