UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

HENRIQUE AWOYAMA KLEIN

Implantação do Suporte a Vídeo Entrelaçado no Módulo de Predição Intra-quadros para o SBTVD

Trabalho de Graduação

Prof. Dr. Altamiro Amadeu Susin
Orientador

Dr. André Borin Soares
Co-orientador

Porto Alegre, julho de 2012

AGRADECIMENTOS

Agradeço a todos colegas da equipe do LaPSI.

Ao Prof. Dr. Altamiro Amadeu Susin pela oportunidade de participar da equipe e pela sua orientação ao longo deste trabalho.

Ao co-orientador Dr. André Borin Soares pela ajuda oferecida diariamente.

Ao colega bolsista Lauro Scheffer Puricelli pela assistência na parte teórica do projeto, e aos demais participantes do grupo envolvido no projeto do Decodificador de vídeo no padrão H.264/MPEG-4 para o SBTVD por proporcionar um ótimo ambiente de trabalho e apoio na realização deste trabalho.

Agradeço também aos meus amigos e a minha família pela motivação, apoio e compreensão durante toda a graduação.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

SBTVD	Sistema Brasileiro de Televisão Digital
LaPSI	Laboratório de Processamento de Sinais e Imagem
DELET	Departamento de Engenharia Elétrica
UFRGS	Universidade Federal do Rio Grande do Sul
ISDB-T	Integrated Services Digital Broadcasting - Terrestrial
DVB	Digital Video Broadcasting
ATSC	Advanced Television Systems Committee
RGB	Espaço de cores “Red, Blue, Green”
YCbCr	Espaço de cores “Luminance, Blue Chrominance, Red Chrominance”
QCIF	Quarter Common Intermediate Format
CABAC	Context-Adaptive Binary Arithmetic Coding
CAVLC	Context-Adaptive Variable Length Coding
DCT	Discrete Cosine Transform
FIFO	First In First Out
FPGA	Field Programmable Gate Array
VHDL	VHSIC Hardware Description Language
VHSIC	Very High-Speed Integrated Circuits
ABNT	Associação Brasileira de Normas Técnicas
PRH.264	Programa de Referência H.264
LUT	Look-Up Table
BRAM	Block Ram

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

Este trabalho está no contexto de decodificação de vídeo no padrão H.264, mais especificamente o módulo de predição Intra-quadros para o SBTVD (Sistema Brasileiro de Televisão Digital). Este possui implementação minimal no projeto, sendo funcional somente no caso mais básico de transmissão, vídeo progressivo. Deste modo é proposta a implantação do funcionamento para vídeo entrelaçado. Para isso inicialmente é descrita uma base teórica para contextualizar o projeto, a situação atual e os requisitos necessários, seguindo da descrição do trabalho que é utilizado como base e as implicações que a implantação do suporte a vídeo entrelaçado traz. Depois disso, é descrita a reimplementação dos módulos de *parsing* e predição Intra-quadros, realizada na linguagem de descrição de *hardware*, VHDL, bem como as ferramentas e metodologias de teste que possibilitaram a validação do projeto. Finalmente, são retiradas conclusões sobre os resultados, incluindo metas futuras e perspectivas sobre o término do projeto.

Palavras-chave: H.264, Predição Intra-quadros, Vídeo entrelaçado, VHDL.

Implantation of the Suport for Interlaced Video in the Intra-frame Prediction Module for the SBTVD

ABSTRACT

This work is within the context of video decoding for the H.264 standard, more specifically the Intra-frame prediction module for the SBTVD (Sistema Brasileiro de Televisão Digital). This module has a minimal implementation in this project, being functional only in the most basic case of video transmission, the progressive video. Thus it is proposed the implantation of the interlaced video functionality. Inicially a theoretical basis is described to contextualize the project, its current situation and requirements, following the description of the work utilized as basis and the implications that the implantation of the suport for interlaced video brings. Thereafter the reimplementation of the *parsing* and Intra-frame prediction modules are described, made in the hardware description language, VHDL, as well as the tools and test methodologies that allowed the validation of the project. Finally, conclusions about the results are drawn, including future goals and prospects on the project completion.

Keywords: H.264, Intra-frame Prediction, Interlaced video, VHDL.

1 INTRODUÇÃO

O projeto do Decodificador de Vídeo no Padrão H.264 teve como seu incentivador o decreto Nº 4.901 em 26 de Novembro de 2003 (DOU, 2003), que instituiu o SBTVD como padrão nacional para Televisão Digital a ser desenvolvido. Assim, para realizar tal tarefa, foram envolvidos vários centros acadêmicos em âmbito nacional, sendo um deles o Laboratório de Processamento de Sinais e Imagem (LaPSI), localizado no Departamento de Engenharia Elétrica (DELET) da Universidade Federal do Rio Grande do Sul (UFRGS). Nesse contexto está sendo desenvolvido do decodificador de vídeo em *hardware*, para decodificação em tempo real, a fim de futuramente permitir o projeto de um chip que realize tal processo.

A motivação para o futuro projeto de um chip que decodifique vídeos nesse padrão vem da necessidade do país de possuir um decodificador produzido em território nacional. Ao passo que o produto é importado, um custo adicional é atrelado, fato que dificulta grande parte da população brasileira de obter acesso à tecnologia. Assim, tendo desenvolvido um chip nacional, a tecnologia proposta no SBTVD estaria em domínio do país, o que resultaria em diminuição do custo do componente e consequentemente maior acessibilidade do povo brasileiro.

O padrão H.264 possui diversas técnicas com o fim de oferecer maior compressão que padrões anteriores. Estas fazem uso do aproveitamento de redundâncias inerentes ao vídeo. Redundâncias temporais se caracterizam pela similaridade entre imagens em instantes de tempo distintos, assim, existe um módulo de predição Inter-quadros, cujo funcionamento é baseado em diminuir a quantidade de informação enviada explorando a característica intrinseca de imagens consecutivas. Redundâncias de frequências exploram a sensibilidade do olho humano, que possibilita desprezar certas faixas de alta frequência cujas amplitudes são baixas sem haver perda de informação significativa, resultado obtível a partir de um agrupamento de frequências espaciais próximas, obtido no módulo de quantização. Redundâncias espaciais dizem respeito à similaridades dentro de uma imagem, portanto existindo um módulo de predição Intra-quadros que aproveita a homogeneidade presente em certas regiões da imagem e reduz a quantidade de dados necessária para repassar a mesma informação. Esta última redundância é o tema deste trabalho.

Atualmente o módulo de predição Intra-quadros está implementado em perfil *baseline*, ou seja, com a sua funcionalidade restrita a casos de vídeo progressivo. Entretanto, de acordo com (ITU-T, 2007), essa configuração diz respeito somente a um dos três tipos de imagem, exemplificados na Tabela 1.1. A imagem *quadro* (frame) está presente em vídeo progressivo. Por sua vez, a imagem *campo* (*field*) é referenciada como *PAFF* (*Picture-Adaptive Frame-Field*) e diz respeito à video entrelaçado em nível de imagem, ou seja, para uma imagem se decide se essa deve ser inteiramente codificada em um quadro ou dois campos (campo *top* e campo *bottom*), um deles contendo as linhas pares e outro as

linhas ímpares. Por fim, pelo fato do padrao H.264 tratar a imagem em blocos de 16x16 pixels (também denominados *macroblocos*), pode-se ter toda a imagem codificada em modo campo ou toda em modo quadro ou ainda pode haver partes da imagem codificadas em campo e partes em quadro. Este último modo é denominado MBAFF (*Macroblock-Adaptive Frame-Field*), e foi trabalhado neste projeto

Tabela 1.1: Tipos de Imagem

Tipo de Imagem	Video entrelaçado	Video progressivo
Imagem *frame*		X
Imagem *field*	X	
Imagem *MBAFF*	X	

Primeiramente no capítulo 2 é apresentada uma introdução básica sobre o contexto do projeto, focando em conceitos, algoritmos e arquiteturas utilizados no âmbito de codificação de vídeo digital. No capítulo 3 é, então, mostrada a implementação do Módulo de Predição Intra-quadros desenvolvido em (BERRIEL, 2005). Seguindo, tem-se no capítulo 4 a Implantação do Suporte do Suporte a Vídeo Entrelaçado, que descreve a contribuição deste projeto, com os resultados descritos no capítulo 5. Finalmente conclusões sobre o trabalho são realizadas no capítulo 6.

2 BASE TEÓRICA

2.1 Sistema Brasileiro de Televisão Digital - SBTVD

O Sistema Brasileiro de Televisão Digital foi criado em 2006 de modo a padronizar a transmissão de televisão digital no país, sendo também adotado em vários outros países da América Latina. Sua inspiração está no padrão japonês ISDB-T (*Integrated Services Digital Broadcasting - Terrestrial*), sofrendo algumas alterações de modo a ser conhecido também como ISDB-Tb (ISDB-T brasileiro). O padrão europeu (DVB - *Digital Video Broadcasting*) e o padrão americano (ATSC *standards - Advanced Television Systems Committee*) foram analisados para serem a base do padrão brasileiro, mas se optou pelo ISDB-T, entre outros motivos, por codificar o vídeo usando H.264 ao invés de MPEG-2, usado pelos outros padrões, provendo cerca de 50% de ganho de eficiência. Em termos de áudio é usado o padrão HE-AAC (*High-Efficiency Advanced Audio Coding*), que possui a mesma qualidade que outros sistemas, utilizando menos banda de frequência. Além destes, outro motivo está no fato do padrão japonês possibilitar o uso de um *middleware*, fornecendo maior interatividade com o usuário.

Até 2016 a transmissão de televisão analógica no país será findada, portanto é necessária a introdução do SBTVD que não somente substituirá a interação com o aparelho televisor, como adicionará funcionalidades. Basicamente o novo padrão fornecerá as seguintes funcionalidades:

- Resolução de vídeo variável até *FULL HD* (1920x1080 pixels).
- Áudio de alta qualidade 5.1.
- Interatividade do usuário com a televisão através de um *middleware*.
- Tolerância a erros de transmissão, evitando “chuviscos” na imagem.
- Multiprogramação, disponibilizando até 3 programas em um mesmo canal.

2.2 Padrão H.264

O padrão de codificação de vídeo H.264 tem como diretiz aumentar ainda mais, em comparação com padrões anteriores, a compressão de dados de modo a possibilitar a transmissão com uma melhor qualidade. Isto se torna necessário pelo fato da largura de banda destinada à transmissão continuar fixa e ao mesmo tempo a qualidade da imagem estar melhorando, deste modo é desejável uma melhor utilização do meio, enviando mais informação e, para isso, utilizando menos dados.

Em comparação com padrões anteriores, o H.264 consegue obter taxas de compressão melhores, podendo em alguns casos chegar até 50% em relação ao MPEG2, 47% em relação ao perfil *baseline* do H. 263 e 23% em relação ao perfil *high*, como visto em (KAMACI, 2007).

Para conseguir tais resultados são utilizados vários módulos que aproveitam a redundância que existe em vídeo, tanto em nível temporal, como espacial e de frequência.

2.2.1 Perfis

No padrão H.264 existem classificações em perfis, de modo a limitar as ferramentas utilizadas em determinados tipos de transmissão de vídeo. Assim, para o SBTVD existem 3 perfis previstos:

- *Baseline*: Perfil mais simples. Não é suportado vídeo entrelaçado, bi-predição e codificador de entropia CABAC. Sua área de utilização é principalmente dispositivos móveis.
- *Main*: Perfil intermediário. Engloba o perfil *baseline* com adição de suporte a vídeo entrelaçado, bi-predição e codificador de entropia CABAC.
- *High*: Perfil mais completo. Possui todas as ferramentas do perfil *main* com possibilidade de utilização de transformadas com blocos de tamanho 8x8 pixels.

2.2.2 Exploração de Redundâncias

A grande compressão obtida por utilizar o padrão H.264 vem da exploração de redundâncias. No caso de uma transmissão de vídeo as redundâncias são categorizadas em três níveis: temporal, espacial e de frequência.

A redundância temporal diz respeito a observar o comportamento do vídeo em função do tempo. Visto que o vídeo é uma sequência de imagens, estas são consecutivas e muitas vezes possuem pouca variação, como podemos ver na Figura 2.1. Assim, é possível diminuir o número de dados enviados se utilizando a redundância entre essas imagens, não necessitando enviar toda imagem seguinte. Usando como parâmetros imagens anteriores e "vetores de movimento", que indicam o sentido do movimento de cada fragmento da imagem, é possível realizar uma predição. Um exemplo pode ser visto na Figura 2.2.

Figura 2.1: Redundância temporal entre imagens em tempos distintos (RICHARDSON, 2003)

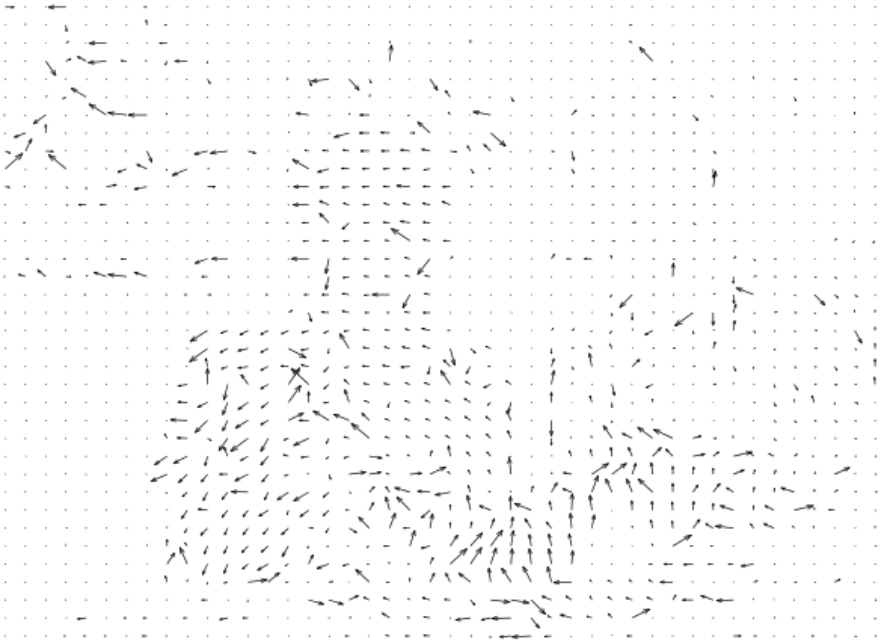

Figura 2.2: Vetores de movimento a partir da Figura 2.1 (RICHARDSON, 2003)

O caso de redundância espacial se caracteriza pela semelhança presente no contexto de uma imagem. Explorando regiões similares é possível se reduzir a quantidade de dados enviados sem perder informação, um exemplo de regiões homogêneas pode ser visto na Figura 2.3. Assim é possível recuperar e informação a partir de pedaços já decodificados da imagem atual, baseando-se em uma média entre os pixels vizinhos.

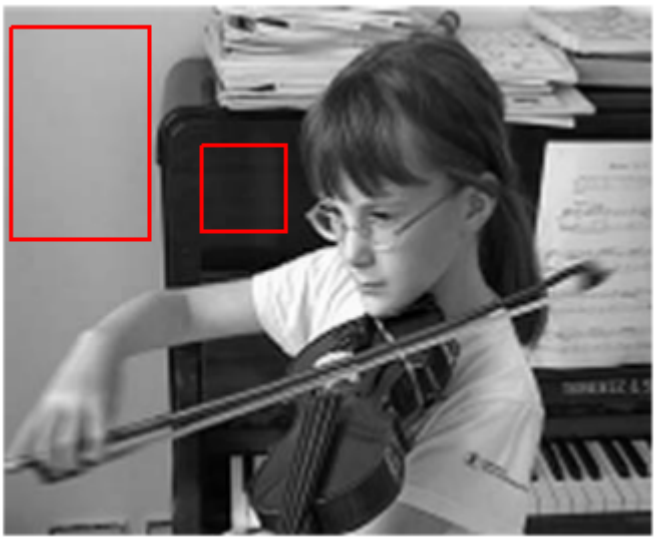

Figura 2.3: Redundância espacial e regiões homogêneas

A redundância de frequência, por sua vez, não é inerente ao vídeo ou à imagem, e sim ao olho humano, que permite diminuir a quantidade de dados pelo fato de ter menos sensibilidade a pequenas amplitudes de faixas de frequências altas. Portanto, para este tipo de redundância existe um módulo de quantização que aumenta a eficiência em detrimento de pouca perda na qualidade da imagem.

2.2.3 Espaço RBG/YCbCr

Para representar uma imagem é necessário um sistema de cores. Esse sistema deve ser capaz de conseguir representar qualquer informação visual por meio de uma composição ponderada de suas componentes. Um sistema comumente utilizado é o RGB (do inglês *Red, Green, Blue*) que consegue representar qualquer imagem usando combinação dessas 3 cores primárias em intensidades diferentes. Entretanto esse sistema não é o mais eficiente, pois ao estudar o comportamento do olho humano é possível se identificar que o mesmo possui muito mais sensibilidade a intensidade da luz que a cor. Com essa motivação foi desenvolvido o sistema de cores YCbCr (do inglês *Y - Luminance, Cb - Blue Chrominance, Cr - Red Chrominance*) que permite "desacoplar"a componente luz das cores. Com essa subtração é necessário enviar somente duas crominâncias e a luminância e com posse dessas é possível então deduzir a terceira crominância. No padrão foi escolhido omitir o envio da crominância verde (Cg) então informando luminância (*Y*), crominância azul (*Cb*) e crominância vermelha (*Cr*). As fórmulas comumente usadas que demonstram como obter um sistema de cores a partir do outro estão presentes nas Equações 2.1 e 2.2. Para maiores referências vide (RICHARDSON, 2003).

$$\begin{aligned} Y &= 0,299R + 0,587G + 0,114B \\ Cb &= 0,564(B - Y) \\ Cr &= 0,713(R - Y) \end{aligned} \tag{2.1}$$

$$\begin{aligned} R &= Y + 1,402Cr \\ G &= Y - 0,344Cb - 0,714Cr \\ B &= Y + 1,772Cb \end{aligned} \tag{2.2}$$

O padrão H.264 utiliza o espaço de cores YCbCr.

2.2.4 Sub-amostragem de cor

Para o espaço de cores YCbCr mostrar de fato alguma vantagem em termos de eficiência é utilizada uma sub-amostragem de cor. No espaço RGB não é possível obter uma sub-amostragem, pois todas as cores precisam da mesma taxa para não ocorrer diferenças na imagem, entretanto, como no YCBCR é subtraída a componente de luminosidade das cores, e tomando posse do fato do olho humano ser mais sensível a luz que a cor, pode-se então diminuir a proporção entre amostragem de cores e luminosidade.

Essa técnica de sub-amostragem permite principalmente três alternativas: 4:4:4, 4:2:2 e 4:2:0. O primeiro dígito indica a proporção de amostras de luminância (Y), o segundo dígito representa a proporção de amostras de crominância azul (Cb), e o último dígito diz a proporção de amostras de crominância vermelha (Cr).

Assim para 4:4:4 temos que todas possuem a mesma proporção e portanto não existe sub-amostragem de fato. Para 4:2:2 temos que a cada 4 amostras de luminância existem 2 amostras de cada crominância atrelada. E finalmente para 4:2:0 é proposto que para cada 4 amostras de luz existem 1 amostra de cada cor, o que a partir da convenção adotada se denominaria “4:1:1”, entretanto 4:2:0 é o nome utilizado por motivos históricos. Na Figura 2.4 estão representadas as diferentes sub-amostragens.

Assim, no padrão H.264, conforme especificado por (ABNT, 2007), se utiliza o espaço YCbCr com sub-amostragem 4:2:0 para maximizar a compressão de dados.

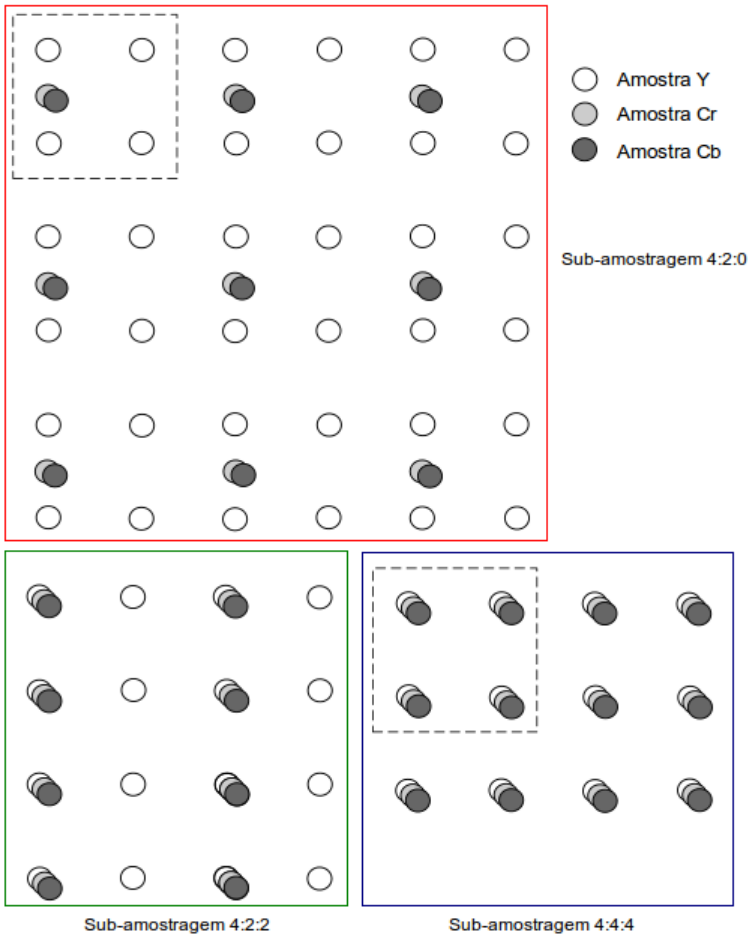

Figura 2.4: Sub-amostragens de cores

2.2.5 Blocos e Macroblocos

Uma imagem no padrão H.264 é dividida em macroblocos e blocos. Macrobloco é o agrupamento de 16x16 pixels de luminância, 8x8 de crominancia azul e 8x8 de crominância vermelha (atendendo a proporção 4:2:0 de sub-amostragem). Já um bloco é uma sub-divisão de um macrobloco, possuindo tamanhos variados. Nas Figuras 2.5 e 2.6 são mostradas as partições, onde o bloco de 4x4 pixels de luminância e 2x2 de cada crominância é a menor posssível.

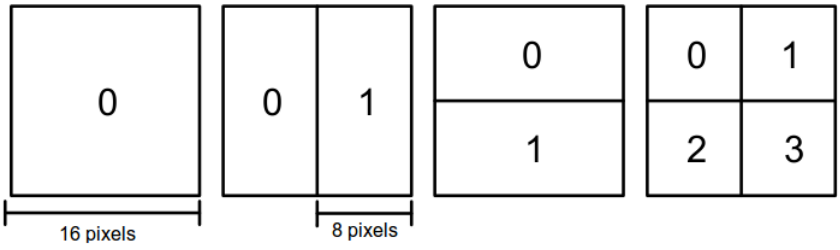

Figura 2.5: Particionamento de um macrobloco

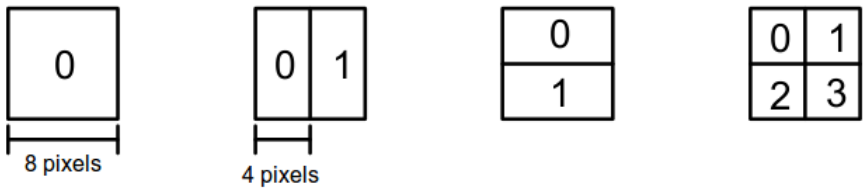

Figura 2.6: Particionamento de um bloco

Cada macrobloco, portanto, possui 4x4 blocos 4x4. A divisão de uma imagem QCIF (*Quarter Common Intermediate Format* - 176x144 pixels) em macroblocos, bem como a ordem dos mesmos pode serem vistos na Figura 2.7.

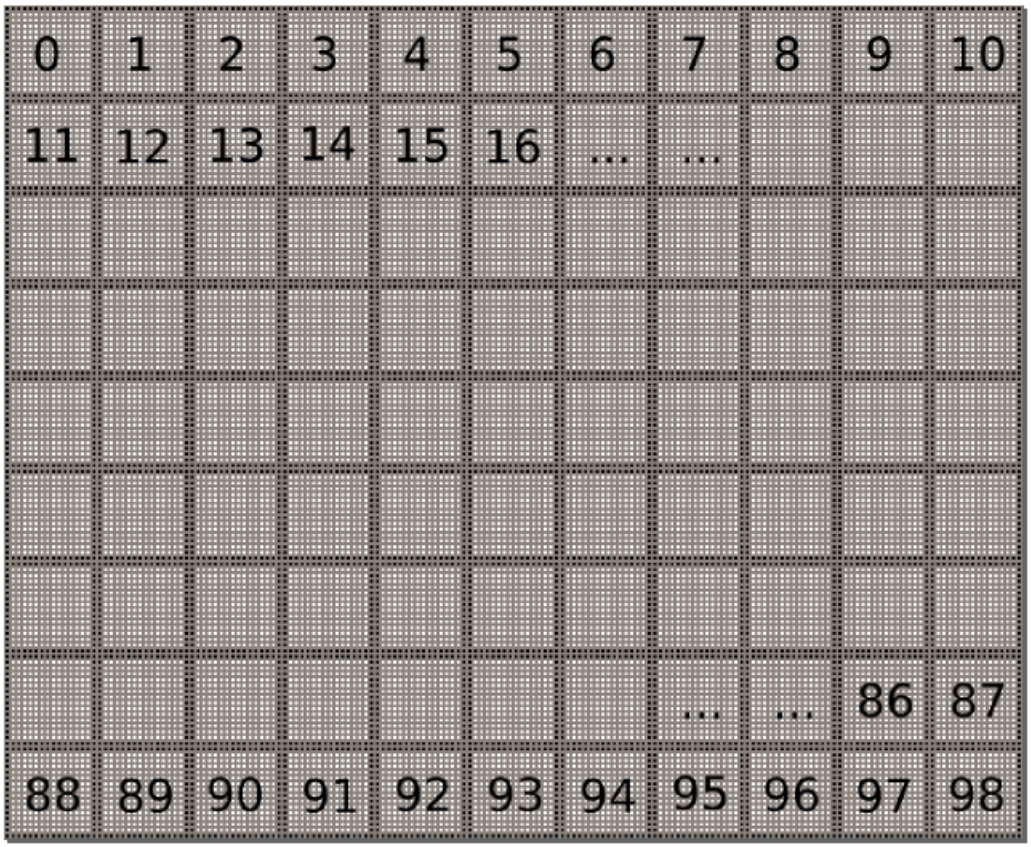

Figura 2.7: Ordem de varredura de macroblocos em uma imagem QCIF (176x144)

Atrelada a essa organização vem também a sua ordem de varredura interna em nível de bloco. Tal ordem está evidenciada na Figura 2.8 e é conhecida como “duplo Z”.

As ordens de varredura e decodificação mencionadas são características de vídeo progressivo. No caso de vídeo entrelaçado ocorrem mudanças que serão especificadas na seção 2.4.

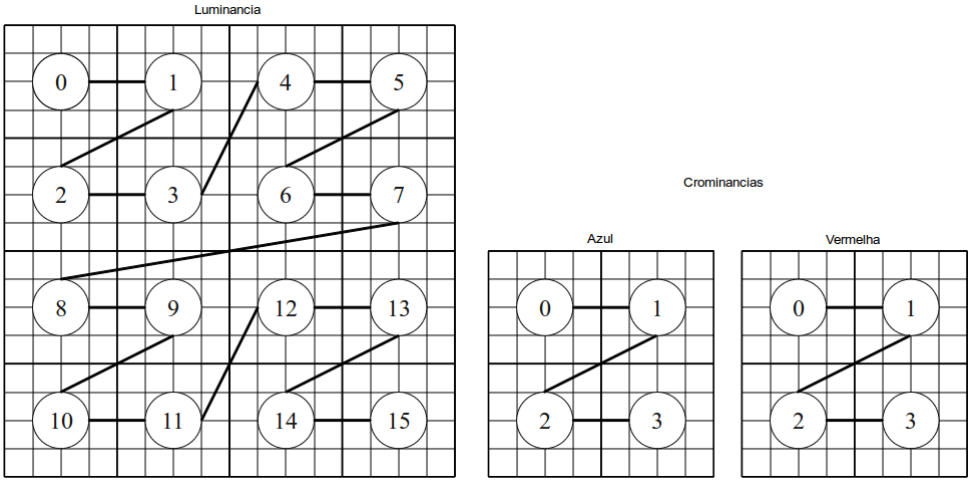

Figura 2.8: Ordem de varredura de amostras de luminância e chrominâncias

2.3 Arquitetura do decodificador H.264

Tendo visto as premissas que o padrão H.264 oferece, a arquitetura do mesmo deve suprir essas funcionalidades e oferecer módulos que organizem o funcionamento da decodificação de vídeo nesse formato de compressão. Assim o modelo da arquitetura de um decodificador H.264 segue a Figura 2.9.

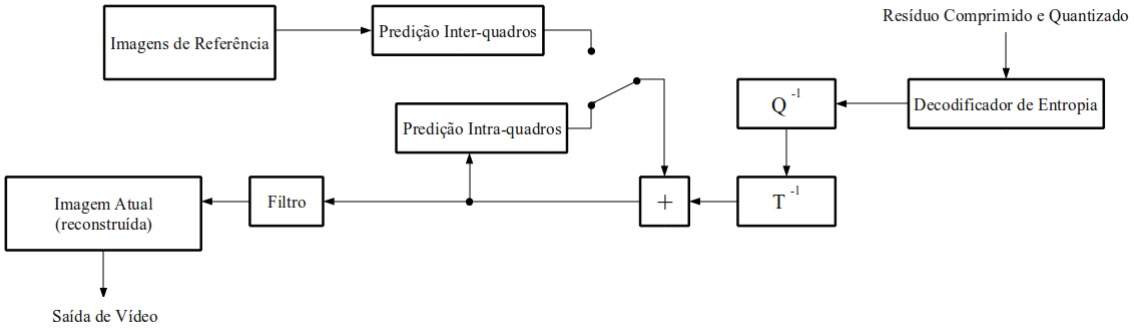

Figura 2.9: Arquitetura do Decodificador H.264

Quando recebidos, os dados passam por uma operação de decodificação de entropia, em seguida descompressão de elementos sintáticos e parâmetros, recuperação dos resíduos e realização de predições. A imagem resultante é a soma do resultado da predição com o resíduo e essa então passa por um filtro de deblocagem. Para os módulos de predição é escolhido o Intra-quadros caso tenha sido usada redundância espacial, ou Inter-quadros caso se tenha explorado redundância temporal.

Os blocos da arquitetura são descritos nas seções seguintes. A seção 2.3.1 descreve um módulo que não está presente na Figura 2.9 por possuir caráter de controle e separação de dados.

2.3.1 Parser

Presente no decodificador está o módulo de parser do *bitstream*. Este serve como controle e interpretação dos pacotes recebidos pelo módulo de forma a identificar parâmet-

ros e configurações do vídeo a ser decodificado e a partir dessas ativar qual dos módulos mostrados na Figura 2.9 deve manipular qual dado. Para maiores detalhes vide (LORENCETTI, 2010).

2.3.2 Decodificador de Entropia

A codificação de entropia tem como base comprimir dados associando símbolos à comprimentos de códigos, de modo que quanto mais frequente um símbolo, menor o código usado para representar este e vice-versa. O trabalho do decodificador de entropia está em reverter esse processo e, como visto na arquitetura presente na Figura 2.9, fornecer coeficientes quantizados de transformada.

No contexto do decodificador H.264 são usados três tipos de codificação de entropia: CABAC (*Context-Adaptive Binary Arithmetic Coding*), CAVLC (*Context-Adaptive Variable Length Coding*) e *Exponential-Golomb*. Os dois primeiros são os codificadores principais, o primeiro possui maior taxa de compressão com custo computacional maior atrelado em comparação com o segundo. O terceiro, por sua vez, é usado em conjunto com um dos outros dois. Maiores detalhes de funcionamento serão omitidos por não fazerem parte do escopo do projeto. Tais detalhes podem ser vistos em (SCHMIDT, 2011).

2.3.3 Quantização e Transformada Inversas

No lado do codificador H.264 são realizadas as operações de transformação e quantização. A primeira tem como meta passar os resíduos do domínio tempo para frequência por meio de uma adaptação da DCT (*Discrete Cosine Transform*) onde são excluídas as multiplicações em ponto flutuante, diminuindo a complexidade computacional. Já a segunda tem como objetivo quantizar os resíduos transformados (denominados "coeficientes") de modo a diminuir a quantidade de informação necessária para representá-los, eliminando os menos significativos e portanto sendo um método de compressão com perdas de informação. Nesse processo que ocorre a exploração da redundância de frequência.

Assim, na parte de quantização e transformadas inversas é realizada a reconstrução do resíduo a partir dos dados obtidos pelo decodificador de entropia.

2.3.4 Predição Inter-quadros

No *bitstream* pode estar indicado o uso de predição Inter-quadro, o que significa que na codificação foi explorada a redundância temporal inerente ao vídeo. Como visto na seção 2.2.2, o funcionamento do módulo se baseia em imagens de referência e vetores de movimento.

Se realiza uma cópia de regiões da referência (anteriormente decodificada) para a região sendo decodificada com deslocamento indicado pelo vetor de movimento com precisão de até um quarto de pixel. É possível se ter duas imagens de referência, de modo que é realizada uma média entre as referências. As imagens utilizadas como referência podem ser do passado ou do futuro, do ponto de vista da ordem de imagens do vídeo. Entretanto, para serem imagens do futuro é necessário que a ordem de decodificação seja alterada. Existem duas listas, "0" e "1", que guardam as imagens de referência do passado e futuro. Quando a predição é realizada a partir de somente uma imagem previamente decodificada, é dito que o bloco atual é um bloco tipo "P". Ao passo que uma predição com duas imagens previamente decodificadas como referência é uma bi-predição, portanto um bloco tipo "B".

Em termos de granularidade, o macrobloco pode ser particionado para ser processado

pelo preditor. Isso permite que para partições diferentes do macrobloco podem ser utilizadas imagens de referência e vetores de movimento diferentes. Dentre as 4 subdivisões possíveis na Figura 2.5, a divisão em 4 frações de 8x8 pode oferecer ainda mais 4 divisões internas, ilustradas na Figura 2.6, estas porém devem possui a mesma imagem de referência, podendo alterar os vetores de movimento.

2.3.5 Predição Intra-quadros

Caso tenha sido escolhida predição Intra-quadros, significa que foi explorada a redundância espacial inerente a cada imagem. Assim, como explicado na seção 2.2.2, o preditor necessita somente da própria imagem, dispensando imagens anteriores ou posteriores.

A predição é realizada a partir de regiões mais próximas do bloco atual. Essas regiões próximas são denominadas "vizinhanças" e estão dispostas à esquerda e acima do bloco a ser predito, como visto na Figura 2.10 para um bloco de 4x4 pixels. O valor do pixel predito é ditado a partir de uma cópia ponderada de seus vizinhos.

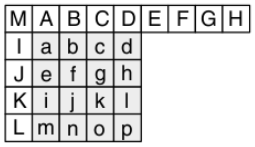

Figura 2.10: Vizinhanças de um bloco 4x4

Para realizar a predição, novamente podem ser utilizados níveis diferentes de granularidade: Macroblocos (16x16 pixels) ou blocos (4x4 pixels). Na Figura 2.11 estão dispostos os modos de predição caso o grão seja um bloco 4x4. Já na Figura 2.12 estão os modos de predição caso o grão seja um bloco 16x16. No caso de crominâncias, como dito na seção 2.2.5, um macrobloco possui 8x8 pixels e para estes deve ser utilizado os mesmos modos 16x16 exceto que nesses casos o modo DC é o "0" e o modo vertical é o "2".

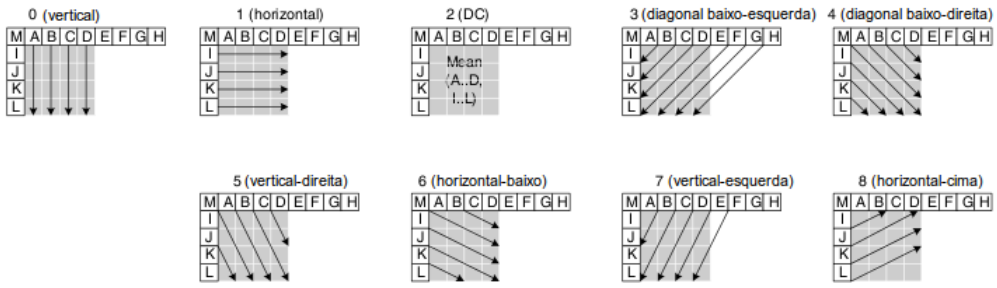

Figura 2.11: Modos de predição Intra-quadros 4x4

Caso venha indicado no *bitstream* que deve ser usado a predição em nível 16x16 também vem informado qual dos modos será utilizado. Caso seja 4x4 é aplicado pelo codificador um algoritmo para derivar os modos de predição de modo a reduzir o máximo possível o resíduo. Este algoritmo vem com um alto custo computacional e é visto com mais detalhes na seção 3.2.1.

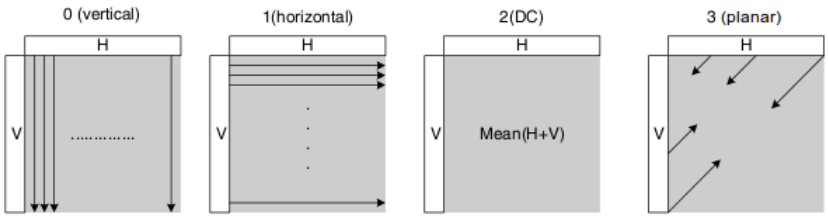

Figura 2.12: Modos de predição Intra-quadros 16x16

2.3.6 Filtro de Deblocagem

Após obter a imagem predita, esta é somada com o resíduo resultante das operações transformada e quantização inversas e assim se obtém uma aproximação boa da imagem final desejada, como podemos ver na Figura 2.13. Entretando ainda existem partes da imagem que possuem distorções causadas pela quantização, e essas podem ser "suavizadas" com a utilização de um módulo de filtragem. No padrão H.264 a filtragem é realizada antes de guardar a imagem como referência, o que aumenta a qualidade da predição Inter-quadros. Na Figura 2.14 está a mesma imagem da Figura 2.13 após a filtragem.

Figura 2.13: Exemplo de imagem não filtrada

Figura 2.14: Mesma imagem da Figura 2.13 após filtragem

2.4 Vídeo Entrelaçado

O vídeo entrelaçado se caracteriza pela diferença na captura do vídeo e na codificação. Ao invés de capturar progressivamente, o vídeo é dividido em linhas pares e ímpares capturadas em instantes de tempo levemente distintos. Isto é, o vídeo possui uma "semi-imagem"contendo as linhas pares e uma outra "semi-imagem"contendo as linhas ímpares, ambas referenciadas como "campos", o campo *top* (de cima) e o campo *bottom* (de baixo). Na Figura 2.15 está disposta a separação dos campos.

Tal modo de codificação permite a diminuição do volume de dados necessários, pois a cada instante é capturada metade da informação que existiria em um vídeo progressivo. Entretanto tal técnica traz um custo em complexidade do decodificador, que como visto em seções anteriores, utiliza técnicas de redundância, as quais devem ser adaptáveis a qualquer tipo de vídeo pré definido pelo padrão H.264. Não somente isto, mas também se adiciona complexidade pelo fato da imagem ter pequenas mudanças nessa diferença de tempos de captura, assim os módulos de predição, filtro, quantização e resíduo devem suprir o correto funcionamento de modo a amortecer e deixar a imagem igual caso esta fosse capturada progressivamente.

Existem 2 níveis de entrelaçamento, em nível de imagem e nível de macrobloco. Entrelaçamento em nível de imagem é conhecido como PAFF (*Picture Adaptive Frame-*

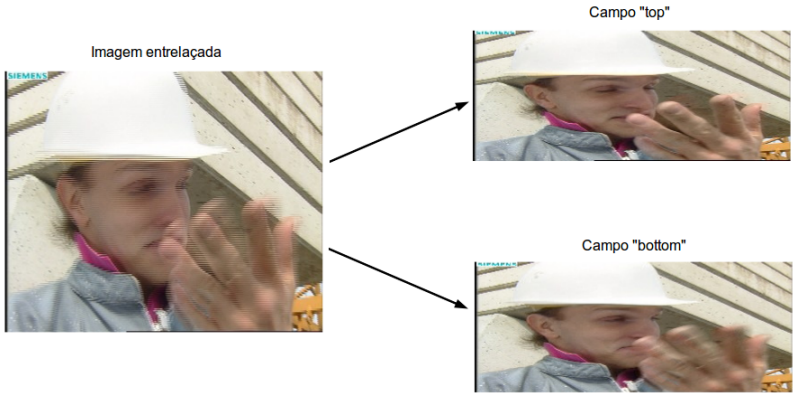

Figura 2.15: Imagem entrelaçada dividida em campos

Field) e em nível de macrobloco é conhecido como MBAFF (*Macroblock Adaptive Frame-Field*). Nas seções 2.4.1 e 2.4.2 é visto o funcionamento de cada técnica.

2.4.1 PAFF

O entrelaçamento adaptativo em nível de imagem decide se vai codificar toda ela em modo *field* (linhas pares e impares separadas em campos *top* e *bottom*) ou em modo *frame* (equivalente ao modo progressivo). Esse tipo de codificação é mais simples no contexto de vídeo entrelaçado e oferece normalmente menor compressão que o MBAFF.

2.4.2 MBAFF

O caso de vídeo entrelaçado adaptativo em nível de macrobloco permite que em uma mesma imagem tenham pares de macroblocos codificados em modo *field* e outros pares em modo *frame*. Assim, o algoritmo de codificação adaptativa pode identificar para cada fração da imagem qual codificação é mais eficiente, caso seja uma região com mais movimento, normalmente os macroblocos são representados em modo *field*, caso contrário, em modo *frame*. A organização da imagem em macroblocos não segue mais a Figura 2.7, e sim uma organização em pares de macroblocos cujo ordenamento segue a Figura 2.16. Cada par de macroblocos possui um macrobloco *top* e um macrobloco *bottom*.

0	2	4	6	8	10	12	...			
1	3	5	7	9	11	...	...			
						...	...	93	95	97
						...	92	94	96	98

Figura 2.16: Ordem de varredura de macroblocos em uma imagem QCIF (176x144) com entrelaçamento MBAFF

Tal ordem acarreta várias mudanças em módulos do decodificador, em termos de vizinhança. A norma (ITU-T, 2007) tem como convenção a determinação de vizinhanças ilustrada na Figura 2.17, entretanto, ao tratar vídeo com MBAFF, a convenção adotada é modificada para o exposto na Figura 2.18.

Figura 2.17: Convenção de vizinhanças em vídeo progressivo

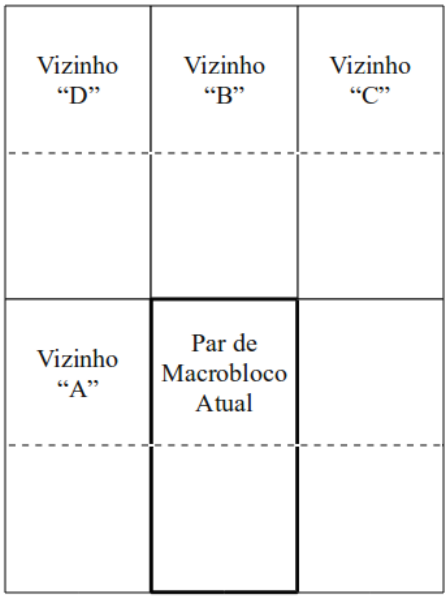

Figura 2.18: Convenção de vizinhanças em vídeo entrelaçado MBAFF

Contudo, somente a mudança de notação não é suficiente para garantir o funcionamento de um decodificador para vídeo entrelaçado. Levando em conta o fato que no interior de uma imagem com MBAFF podem existir macroblocos tipo *frame* e macroblocos tipo *field* existem combinações diferentes de vizinhanças e , para isto, os vizinhos corretos para cada caso estão indicados a partir da Tabela 2.1.

As implicações dessas mudanças nos módulos do decodificador serão vistas no capítulo 4.

Tabela 2.1: Tabela de Vizinhanças para vídeo com MBAFF

Vizinhança	Tipo de Codificação do Macrobloco Atual	Identificador do Par	Tipo de Codificação do Macrobloco Vizinho	Informação Adicional	Atribuição do Vizinho	Caso
Lateral	Frame	Top	Frame		Lateral <= Vizinho “top” A	0
			Field	Índice do pixel é par	Lateral <= Vizinho “top” A	1
				Índice do pixel é ímpar	Lateral <= Vizinho “bottom” A	2
		Bottom	Frame		Lateral <= Vizinho “bottom” A	3
			Field	Índice do pixel é par	Lateral <= Vizinho “top” A	4
				Índice do pixel é ímpar	Lateral <= Vizinho “bottom” A	5
	Field	Top	Frame	Índice do pixel é menor que metade da altura do macrobloco	Lateral <= Vizinho “top” A	6
				Índice do pixel é maior ou igual a metade da altura do macrobloco	Lateral <= Vizinho “bottom” A	7
			Field		Lateral <= Vizinho “top” A	8
		Bottom	Frame	Índice do pixel é menor que metade da altura do macrobloco	Lateral <= Vizinho “top” A	9
				Índice do pixel é maior ou igual a metade da altura do macrobloco	Lateral <= Vizinho “bottom” A	10
			Field		Lateral <= Vizinho “bottom” A	11
Superior	Frame	Top	Frame ou Field		Superior <= Vizinho “bottom” B	12
		Bottom	Frame ou Field		Superior <= “top” Atual	13
	Field	Top	Frame		Superior <= Vizinho “bottom” B	14
			Field		Superior <= Vizinho “top” B	16
		Bottom	Frame ou Field		Superior <= Vizinho “bottom” B	17
Diagonal Esquerda	Frame	Top	Frame ou Field		Diagonal Esquerda <= Vizinho “bottom” D	18
		Bottom	Frame	Vizinho de referência é o “A”	Diagonal Esquerda <= Vizinho “top” A	19
			Field	Vizinho de referência é o “A”	Diagonal Esquerda <= Vizinho “bottom” A	20
	Field	Top	Frame		Diagonal Esquerda <= Vizinho “bottom” D	21
			Field		Diagonal Esquerda <= Vizinho “top” D	22
		Bottom	Frame ou Field		Diagonal Esquerda <= Vizinho “bottom” D	23
Diagonal Direita	Frame	Top	Frame ou Field		Diagonal Direita <= Vizinho “bottom” C	24
		Bottom	não disponível	vizinho não disponível	Não há vizinhança Diagonal Direita	25
	Field	Top	Frame		Diagonal Direita <= Vizinho “bottom” C	26
			Field		Diagonal Direita <= Vizinho “top” C	27
		Bottom	Frame ou Field		Diagonal Direita <= Vizinho “bottom” C	28

3 MÓDULO DE PREDIÇÃO INTRA-QUADROS

O módulo de predição Intra-quadros possui implementação em VHDL (*VHSIC Hardware Description Language*) com perfil *baseline*, ou seja, não suporta predição de vídeos que contenham entrelaçamento. Assim, nesse capítulo, é mostrado o algoritmo realizado, bem como a arquitetura escolhida em (BERRIEL, 2005).

3.1 Algoritmo Implementado

Como visto na seção 2.3.5, o funcionamento do preditor Intra-quadros não necessita de imagens de referência. Através das vizinhanças do bloco (ou macrobloco) atual e dos modos de predição é possível obter a saída desejada. Após esse processo, então, é somado o resíduo à predição, para este poder ser utilizado como referência sem propagação de erros. Também é necessário identificar qual tipo de predição é realizada: 16x16 ou 4x4. A predição 8x8 de crominâncias possui funcionamento similar ao 16x16 de luminância, por ser tratada no nível de macrobloco.

Estruturando as operações descritas, um algoritmo do módulo pode ser visto na Figura 3.1. Primeiramente é recebido o contexto do macrobloco, que indica informações gerais do mesmo, como: localização dentro da imagem, largura da imagem, sub-amostragem de cor, modos de predição e outros dados que serão explicados em mais detalhes na seção 3.2.1.

Com as informações do macrobloco disponíveis é identificado qual o tipo de predição que deve ser utilizado (blocos 4x4 ou 16x16), então é verificada a disponibilidade dos vizinhos (ex: Se o macrobloco pertence à primeira linha então não possui vizinho superior), e finalmente derivam-se os modos de predição. Tendo essas informações em mãos o módulo deve buscar as vizinhanças necessárias e então gerar a predição. Após isto, o resultado da predição deve ser adicionado ao resíduo e então armazenado. Finalmente se verifica qual tipo de predição foi realizado, caso seja em nível de macrobloco (16x16) então deve-se receber um novo contexto do próximo macrobloco, caso seja em nível de bloco (4x4) o módulo deve verificar se o bloco atual é o último do macrobloco atual, se é o caso então a predição para o macrobloco terminou, caso contrário gera a predição do próximo bloco 4x4.

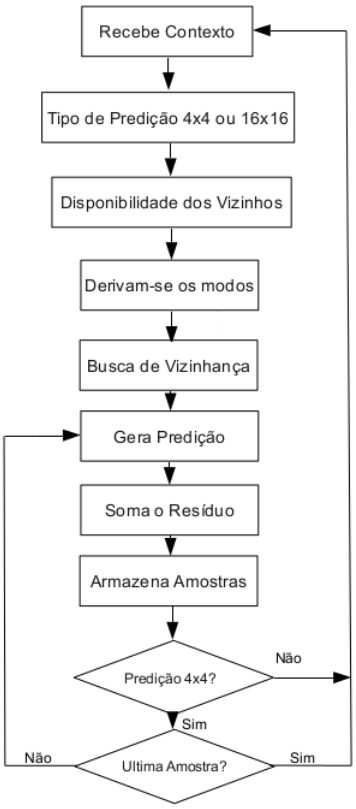

Figura 3.1: Algoritmo de funcionamento do módulo de predição Intra-quadros

3.2 Arquitetura do Preditor Intra-quadros

A partir do algoritmo criado foi implementada a arquitetura do módulo de predição Intra-quadros, presente na Figura 3.2. A subdivisão lógica por funcionalidade está indicada pela legenda. Estes quatro blocos funcionais são descritos na seções 3.2.1, 3.2.2, 3.2.3 e 3.2.4.

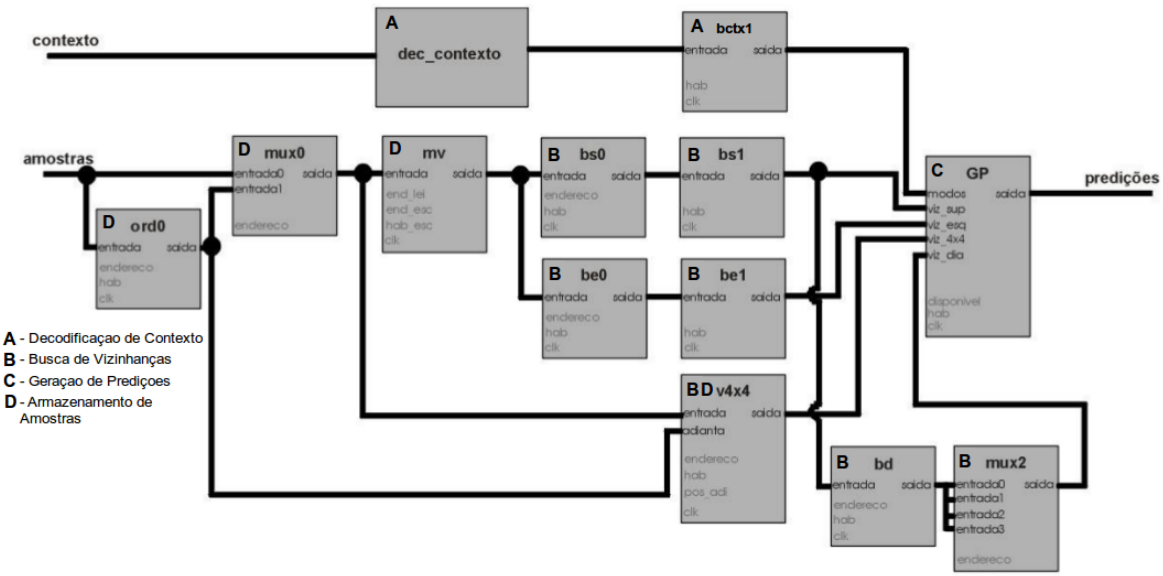

Figura 3.2: Diagrama de blocos da arquitetura do preditor Intra-quadros

3.2.1 Decodificação de Contexto

O *contexto* contém um conjunto de informações necessárias para predição do macrobloco. Deste modo, para cada macrobloco existe um contexto atrelado. A largura de cada *contexto* consiste em 108 bits e as informações que estão contidas no mesmo podem ser vistas na Tabela 3.1.

Tabela 3.1: Informações contidas no *contexto*

Informação	Nº de bits	Posição no vetor	Descrição
col	8	107-100	Posição do macrobloco na linha atual.
addr	16	99-84	Posição do macrobloco na imagem atual.
larg	8	83-76	Largura da imagem em macroblocos.
cor	2	75-74	Sub-amostragem de cor.
tipo_modo	5	73-69	Tipo de predição. Caso seja 16x16 informa o modo.
fmodo	16	68-53	16 Indicadores de quais blocos 4x4 tem o modo derivado por predição.
vmodo	48	52-5	16 Modos , que variam de 0 a 7, dos blocos 4x4 que não tem modo derivado por predição.
cmodo	2	4-3	Modo de predição das crominâncias.
nao_usar	1	2	Indicador se é possível usar o macrobloco como vizinho para predições Intra-quadros.
ff	2	1-0	Espaço reservado para indicadores de uso de vídeo entrelaçado. (não implementado)

Assim, a partir da decodificação dessa estrutura de dados, é possível informar a disponibilidade dos vizinhos, o tipo de predição (16x16 ou 4x4) e os modos de predição atrelados, que são depositados em outra estrutura de dados chamada *ctx* que contém 76 bits organizados de acordo com a Tabela 3.2. Após a decodificação do contexto, a saída é chaveada pelo módulo *bctx1* e encaminhada ao módulo de geração de predições.

Entre todas as informações decodificadas, as que requerem um algoritmo mais complexo são os modos de predição 4x4, pois, como visto na seção 2.3.5, não vêm indicados pelo *bitstream* ao contrário dos modos 16x16. Para deduzir o modo de predição adequado são enviados dois dados definidos pela norma (ITU-T, 2007) e a partir destes, em conjunto com as informações internas do módulo, deve-se derivar os modos de predição. Os dados vindos no *bitstream* são denominados *prev_intra4x4_pred_mode_flag* e *rem_intra4x4_pred_mode*, que indicam se o modo calculado deve ser utilizado e todos os modos possíveis quando não há cálculo de modos, respectivamente. Então, tomando posse do conhecimento que em um macrobloco 16x16 existem 16 blocos 4x4, e assim numerando-os de “0” a “15”, considerando “A” o vizinho à esquerda e “B” o vizinho superior, temos o algortimo ilustrado na Figura 3.3.

Os dados provindos do codificador são obtidos, em geral, a partir de um algoritmo que visa maximizar a compactação do *bitstream*, tal como o *SAD* (*Sum of Absolute Differences*).

Tabela 3.2: Informações contidas na saída *ctx*

Informação	Nº de bits	Posição no vetor	Descrição
disp_esq	1	75	Indicador da disponibilidade do vizinho esquerdo.
disp_sup	1	74	Indicador da disponibilidade do vizinho superior.
disp_sd	1	73	Indicador da disponibilidade do vizinho diagonal direito.
disp_se	1	72	Indicador da disponibilidade do vizinho diagonal esquerdo.
não_usar	1	71	Indicador se é possível usar o macrobloco como vizinho para predições Intra-quadros.
reservado	1	70	Reservado para possíveis modificações.
tipo	2	69-68	Tipo da predição: 16x16, 4x4 ou "não é intra".
modochroma	2	67-66	Modo de predição das crominâncias.
modo16x16	2	65-64	Modo de predição 16x16
modo4x4_0	4	63-60	Modo de predição 4x4 do bloco 0
modo4x4_1	4	59-56	Modo de predição 4x4 do bloco 1
modo4x4_2	4	55-52	Modo de predição 4x4 do bloco 2
modo4x4_3	4	51-48	Modo de predição 4x4 do bloco 3
modo4x4_4	4	47-44	Modo de predição 4x4 do bloco 4
modo4x4_5	4	43-40	Modo de predição 4x4 do bloco 5
modo4x4_6	4	39-36	Modo de predição 4x4 do bloco 6
modo4x4_7	4	35-32	Modo de predição 4x4 do bloco 7
modo4x4_8	4	31-28	Modo de predição 4x4 do bloco 8
modo4x4_9	4	27-24	Modo de predição 4x4 do bloco 9
modo4x4_10	4	23-20	Modo de predição 4x4 do bloco 10
modo4x4_11	4	19-16	Modo de predição 4x4 do bloco 11
modo4x4_12	4	15-12	Modo de predição 4x4 do bloco 12
modo4x4_13	4	11-8	Modo de predição 4x4 do bloco 13
modo4x4_14	4	7-4	Modo de predição 4x4 do bloco 14
modo4x4_15	4	3-0	Modo de predição 4x4 do bloco 15

3.2.2 Busca de Vizinhanças

A busca de vizinhanças tem o papel de suprir o módulo de geração de predições com as amostras vizinhas do bloco/macrobloco. Em termos de vizinhança 16x16 é necessário buscar na memória de vizinhanças (*mem_viz*) dois tipos de vizinhança 16x16: superior e lateral. Os módulos *bs0* e *be0* armazenam as 16 amostras vizinhas superiores e laterais, respectivamente. Já em termos de amostras diagonais, é encaminhada para o módulo *bd* a amostra da diagonal esquerda, visto que não é necessário toda a linha inferior. A vizinhança 16x16 buscada na memória para cada macrobloco está ilustrada na Figura 3.4. Os blocos *bs1* e *be1* tem como função chavear as amostras de seus módulos precedentes.

```
Para BlockIndex entre 0 e 15:

        Se "A" não está disponível ou não é 4x4:
                modoA = 2;
        Senão
                modoA = modo de predição do bloco "A";

        Se "B" não está disponível ou não é 4x4:
                modoB = 2;
        Senão
                modoB = modo de predição do bloco "B";

        pmodo = menor valor entre modoA e modoB;

        Se pred_intra4x4_pred_mode_flag(BlockIndex) = 1:
                modo(BlockIndex) = pmodo;
        Senão
                Se rem_intra4x4_pred_mode(BlockIndex) < pmodo:
                        modo(BlockIndex) = rem_intra4x4_pred_mode_flag(Block_Index);
                Senão
                        modo(BlockIndex) = rem_intra4x4_pred_mode_flag(Block_Index) + 1;

Repete para cada bloco;
```

Figura 3.3: Algoritmo para calcular os modos de predição 4x4

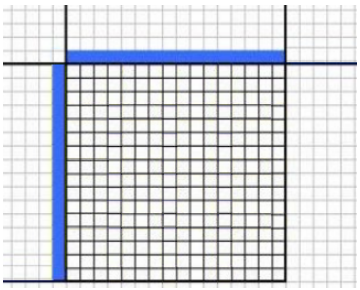

Figura 3.4: Amostras vizinhas 16x16 buscadas da memória para um macrobloco

A outra funcionalidade da busca de vizinhanças é de suprir a vizinhança 4x4 para o gerador de predição. Visto que a predição 4x4 tem como vizinhas as amostras do macrobloco sendo atualmente predito é necessário eliminar o *overhead* de passar as amostras por uma memória antes de utilizá-las. Assim foi construído o módulo *v4x4* a partir de registradores a fim de encaminhar as vizinhanças 4x4.

Detalhes sobre a organização da memória e dos registradores dedicados às amostras vizinhas são descritos na seção 3.2.4.

3.2.3 Geração de Predições

O gerador de predições possui dois módulos internos: GP_4x4 e GP_Luma16, que realizam as predições 4x4 e luminância 16x16, respectivamente. Por escolha do desenvolvedor, foi construído de modo a gerar 4 amostras por ciclo de execução. Possui uma latência de 4 ciclos até o início da predição e segue a ordem de amostras ilustrada na Figura 3.5.

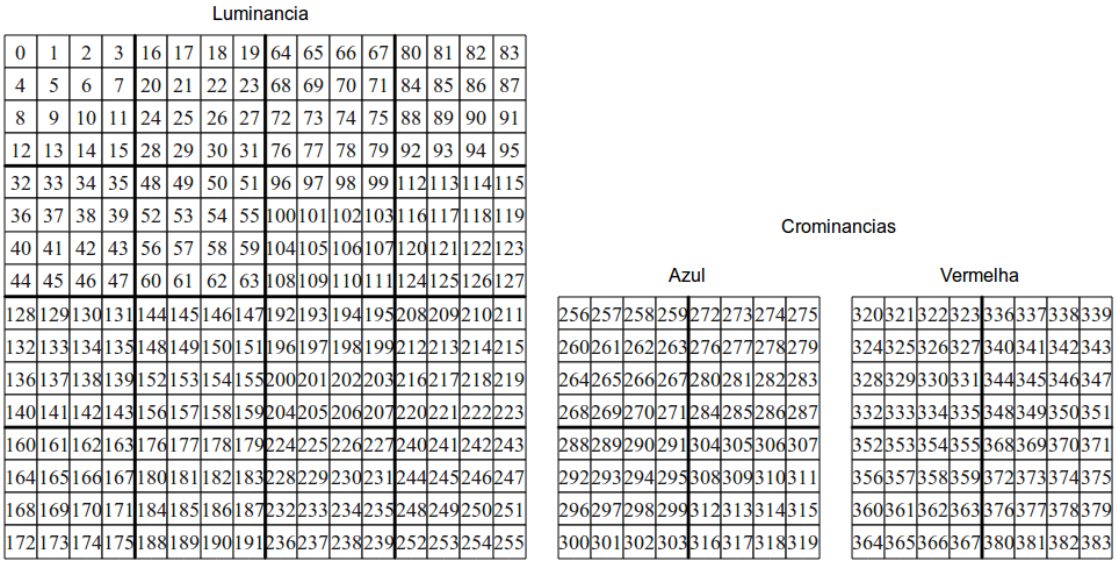

Figura 3.5: Ordem de geração de amostras preditas

3.2.4 Armazenamento de Amostras

Tendo terminado o ciclo de predição de uma amostra, a mesma é somada com o resíduo fora do módulo de predição Intra-quadros e realimentada na entrada do mesmo para ser armazenada na memória de vizinhança (*mem_viz*) ou, no caso de predição 4x4, encaminhada para registradores para uso mais imediato (*v4x4*).

A memória de vizinhança possui capacidade para armazenar um número de vizinhos superiores igual a largura da imagem e uma coluna de vizinhos laterais, como ilustrado na Figura 3.6. A organização da memória segue a Tabela 3.3.

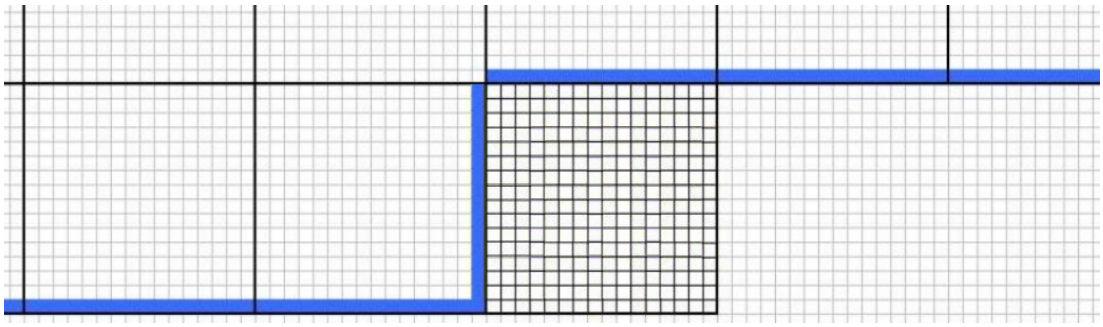

Figura 3.6: Amostras vizinhas 16x16 armazenadas em memória

Os vizinhos diagonais são guardados no módulo *bd* em três registradores, para uma amostra de luminância e uma de cada crominância. Já os vizinhos internos ao macrobloco, ou seja, para predição 4x4, que necessitam ser registrados, são 84 amostras destacadas em amarelo na Figura 3.7.

Tabela 3.3: Organização da memória de vizinhanças

Posição de memória	Nº de amostras	Tipo de vizinhança
0-479	1920	Linha de vizinhos superiores de luminância.
504-507	16	Coluna de vizinhos laterais de luminância.
508-509	8	Coluna de vizinhos laterais de crominância azul.
510-511	8	Coluna de vizinhos laterais de crominância vermelha.
512-751	960	Linha de vizinhos superiores de crominância azul.
768-1007	960	Linha de vizinhos superiores de crominância vermelha.

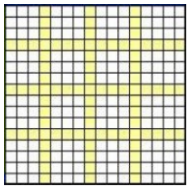

Figura 3.7: Registradores necessários para vizinhança 4x4

Contudo, se valendo do fato que não é necessário que em todo tempo de predição 4x4 todas as 84 amostras estejam disponíveis, o numéro de registradores é reduzido para 28. Fato demonstrado a partir da execução de cada bloco, esquematizada na Figura 3.8.

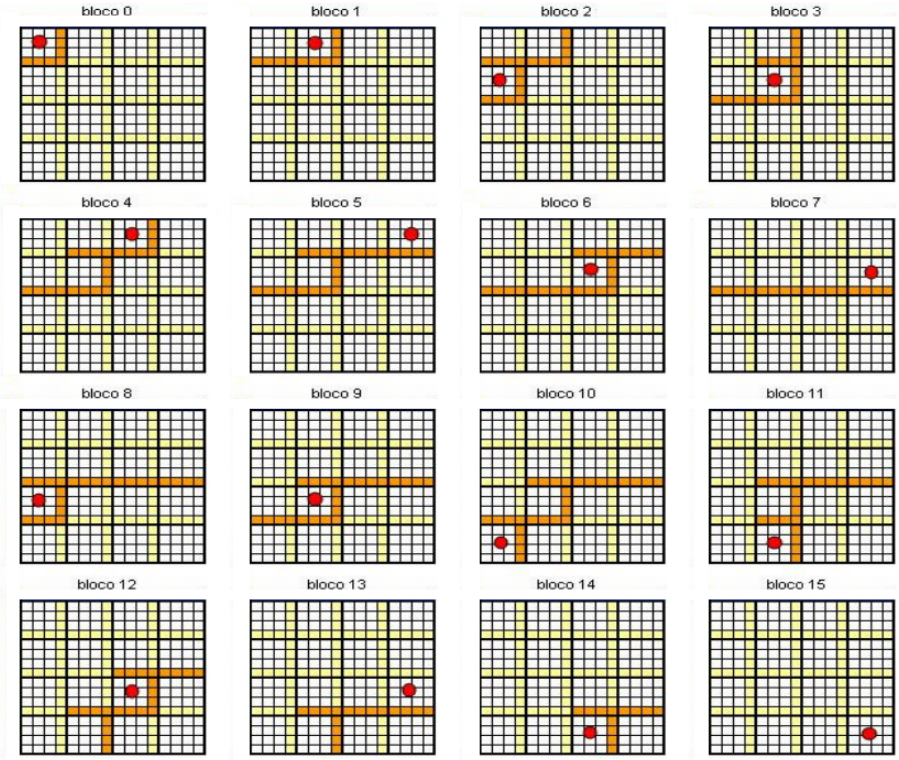

Figura 3.8: Registradores necessários para cada bloco, otimizando o número máximo de registradores para 28

4 IMPLANTAÇÃO DO SUPORTE A VÍDEO ENTRELAÇADO

Tomando por base o trabalho implementado em (BERRIEL, 2005), descrito no capítulo 3, foi então proposta a adição de suporte a vídeo entrelaçado. Assim, foram reformulados e reimplementados elementos da arquitetura do projeto do decodificador H.264.

Primeiramente foram analisadas as entradas necessárias para o novo módulo. A análise de requisitos retornou que dois dados necessitavam serem obtidos a partir de módulos exteriores:

- *Indicador de uso de MBAFF na imagem*
- *Indicador de tipo de macrobloco: field ou frame*

Essas informações são obtidas através da operação de *parsing* do *bitstream* realizada no módulo *parser*, brevemente descrito na seção 2.3.1. Portanto foi estudada a estrutura do parser e observado o modo de encapsulamento das informações utilizadas para formar o *contexto* do módulo de predição Intra-quadros.

Na seção 4.1 são vistas as estruturas envolvidas em tal processo, modificações necessárias e reimplementações para tornar possível a adição de suporte a vídeo entrelaçado no módulo de predição Intra-quadros.

4.1 Implantação do Suporte no Parser

4.1.1 Propagação de *Flags* de Entrelaçamento

O *parser* do decodificador H.264, implementado em (LORENCETTI, 2010), obtém os indicadores necessários de entrelaçamento, entretanto precisavam ser encapsulados e propagados em sincronismo com as outras informações de predição ao módulo destino. Para comunicar o parser com o preditor uma FIFO (*First In First Out*) é utilizada de modo a guardar os dados de predição. Após o encapsulamento dos dados no *parser*, a estrutura é depositada na FIFO e retirada pelo módulo de predição.

De acordo com a norma (ITU-T, 2007) as *flags* que indicam uso de MBAFF e tipo de macrobloco são denominadas respectivamente *MBaffFrameFlag* e *mb_field_decoding_flag*. Deste modo estes sinais foram retiradas dos módulos *slice header* e *slice data* e alocados na estrutura de dados direcionada à FIFO de predição. A arquitetura do *parser*, bem como os módulos alterados, podem ser vistos na Figura 4.1.

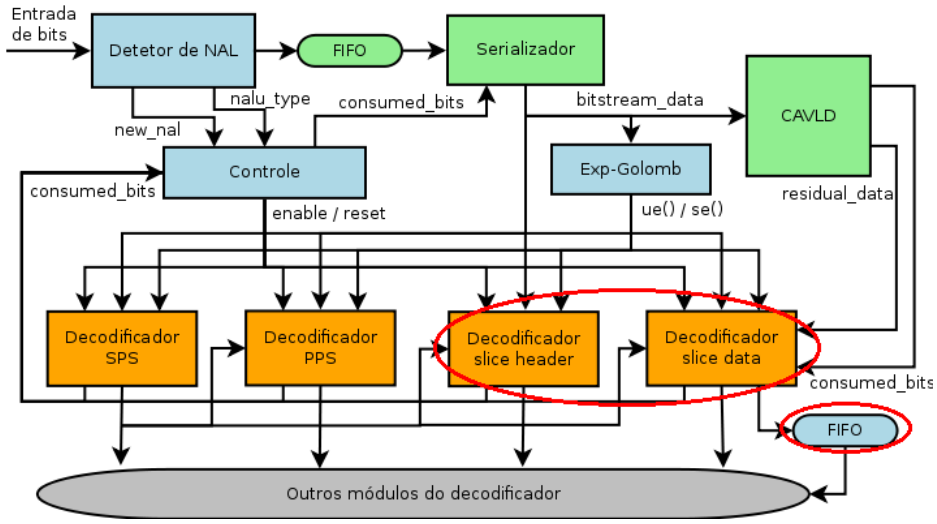

Figura 4.1: Arquitetura do *parser* (LORENCETTI, 2010)

Portanto, tendo as *flags* devidamente sincronizadas o parser foi testado para um caso de MBAFF cujas imagens são todas entrelaçadas, isto é, ambas as *flags* de entrelaçamento possuem valor "1". O resultado obtido apresentou erros nos sinais mencionados e através de uma contagem de número de bits consumidos por cada componente do parser, foi evidenciado que o módulo de cálculo de coeficientes não nulos de cada macrobloco não estava de acordo com a norma (ITU-T, 2007) e necessitava reimplementação.

4.1.2 Reimplementação do Módulo de Cálculo de Coeficientes Não Nulos

O módulo *nc*, que realiza o cálculo dos coeficientes não nulos, tem como função realizar uma estimativa do número de coeficientes não nulos ("nc") de blocos da imagem e a partir destes indicar quantos bits devem ser consumidos do *bitstream* e após isso ajustar o valor do "nc" do bloco a partir do obtido no *bitstream*.

Para estimar o "nc" de cada bloco é realizada uma média dos "ncs" dos vizinhos superior e esquerdo do bloco. A implementação original do módulo *nc* possui uma memória interna de modo a armazenar uma linha de vizinhos superiores e não possui nenhuma estrutura dedicada a guardar os vizinhos da esquerda, deixando-as em sinais internos, pois o modo de varredura de vídeo progressivo (Figura 2.7) faz com que não seja necessário manter os dados, visto que na próxima execução os últimos "ncs" calculados do bloco anterior estarão ainda nos sinais internos e coincidem sempre com a vizinhança da esquerda. Tal organização é ilustrada na Figura 4.2.

Como visto na seção 2.4.2, a ordem de varredura é em pares de macroblocos (vide Figura 2.16). Então analogamente ao realizado em vídeo progressivo, a nova ordem de varredura foi aproveitada na construção da arquitetura entrelaçada. A partir da análise das possibilidades da Tabela 2.1, é visto que é necessário armazenar duas linhas de vizinhanças superiores, e duas colunas de vizinhanças laterais. Para possibilitar mudanças em nível de pixel a pixel (em caso de vizinhança lateral) foi implementada a arquitetura da Figura 4.3. Com esta escolha, é possível se aproveitar da ordem de decodificação nova, que varre em nível de pares de macroblocos, se adicionando o dobro de registradores e o dobro de memória, endereçados com os marcadores "*TOP*" e "*BOTTOM*".

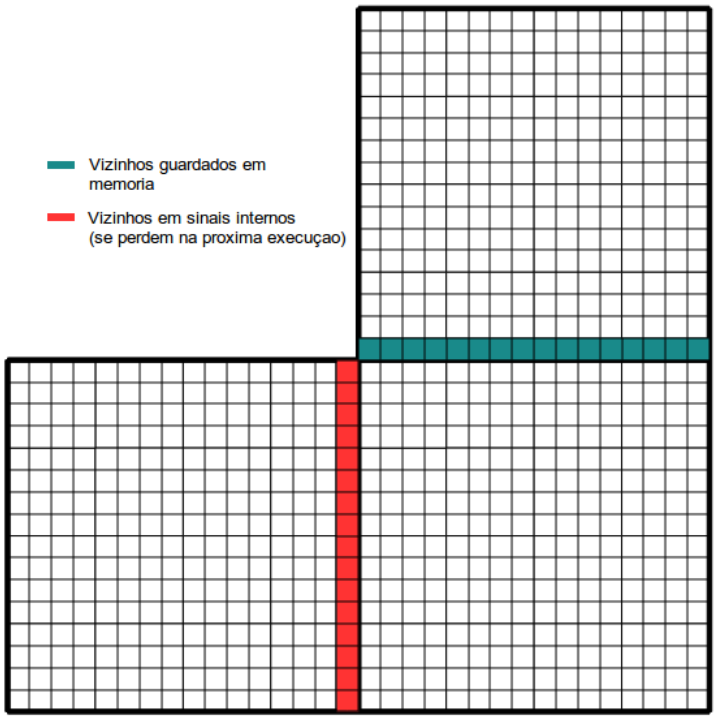

Figura 4.2: Organização das vizinhanças do módulo *nc*

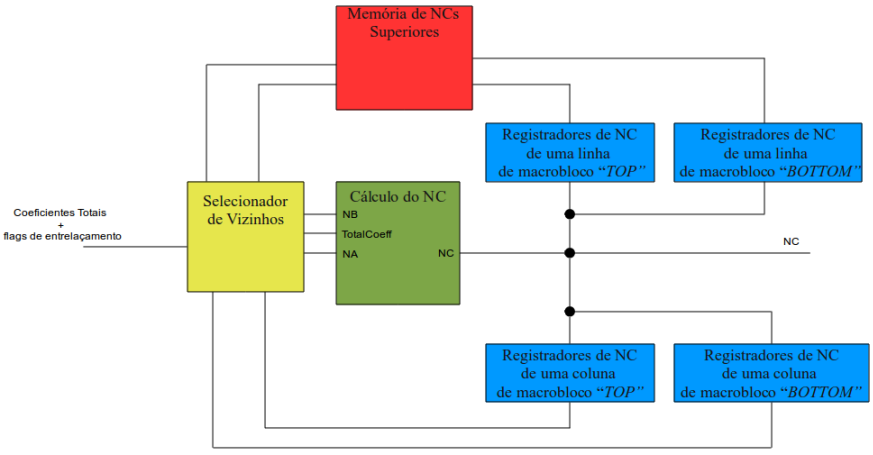

Figura 4.3: Nova arquitetura do módulo *nc*

Com esta reimplementação, é possível ter funcionamento para qualquer caso de vídeo no *parser*, pois agora a organização de amostras disponíveis da vizinhança segue a Figura 4.4, que possibilita qualquer caso da Tabela 2.1.

Portanto, após verificado o funcionamento correto do *parser* foi possível dar início às implementações inicialmente propostas no projeto.

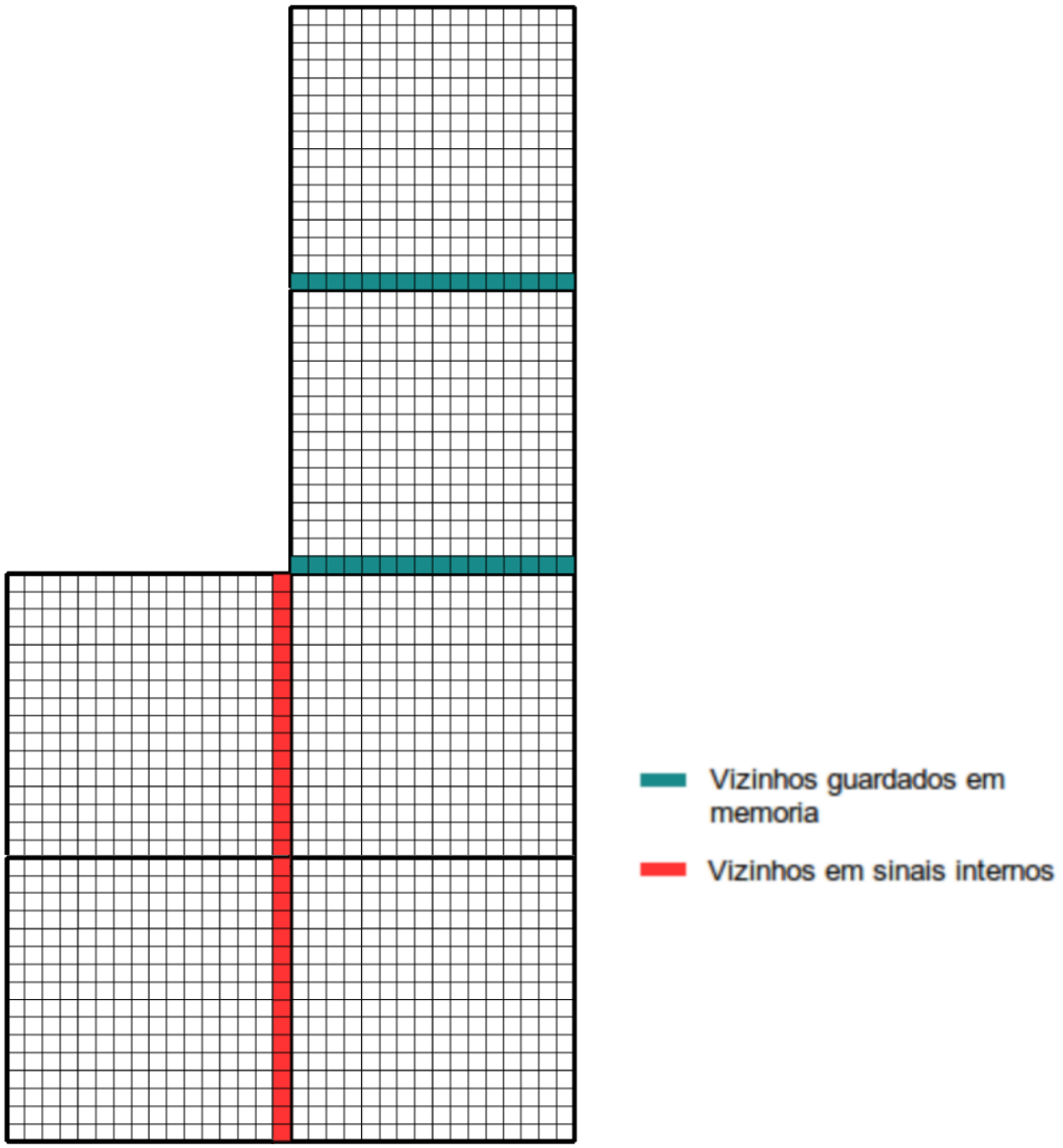

Figura 4.4: Nova organização das vizinhanças do módulo *nc*

4.2 Implantação do Suporte no Preditor Intra-quadros

Após a propagação das informações de entrelaçamento serem efetivadas em outros módulos, foi possível reestruturar o módulo de predição Intra-quadros para funcionar também para vídeo entrelaçado. A análise de requisitos resultou que as reimplementações necessárias envolvem três funcionalidades:

- Ajuste do *contexto* para MBAFF
- Adaptação do armazenamento das amostras
- Modificação na busca de vizinhanças

O módulo funcional de geração de predições, descrito na seção 3.2.3, não possui mudanças, visto que os algortimos de predição não modificam em caso de MBAFF.

4.2.1 Ajuste de Contexto

Na operação inicial o *contexto* é gerado a partir da extração das informações contidas na FIFO de predição. Como visto anteriormente na seção 3.2.1, Tabela 3.1, existem dois bits reservados para o uso de entrelaçamento e nestes são colocados as *flags* de vídeo entrelaçado. Apesar das informações de predição serem oriundas do parser, as informações de posição do macrobloco na imagem e na linha são definidas no próprio módulo topo do preditor. Adicionalmente existe o indicador de *top* ou *bottom* nomeado neste módulo como *CurrMbIsTop*. Para colocar essas modificações foi incluído um sinal mais significativo no *contexto* e seguido o algoritmo presente na Figura 4.5.

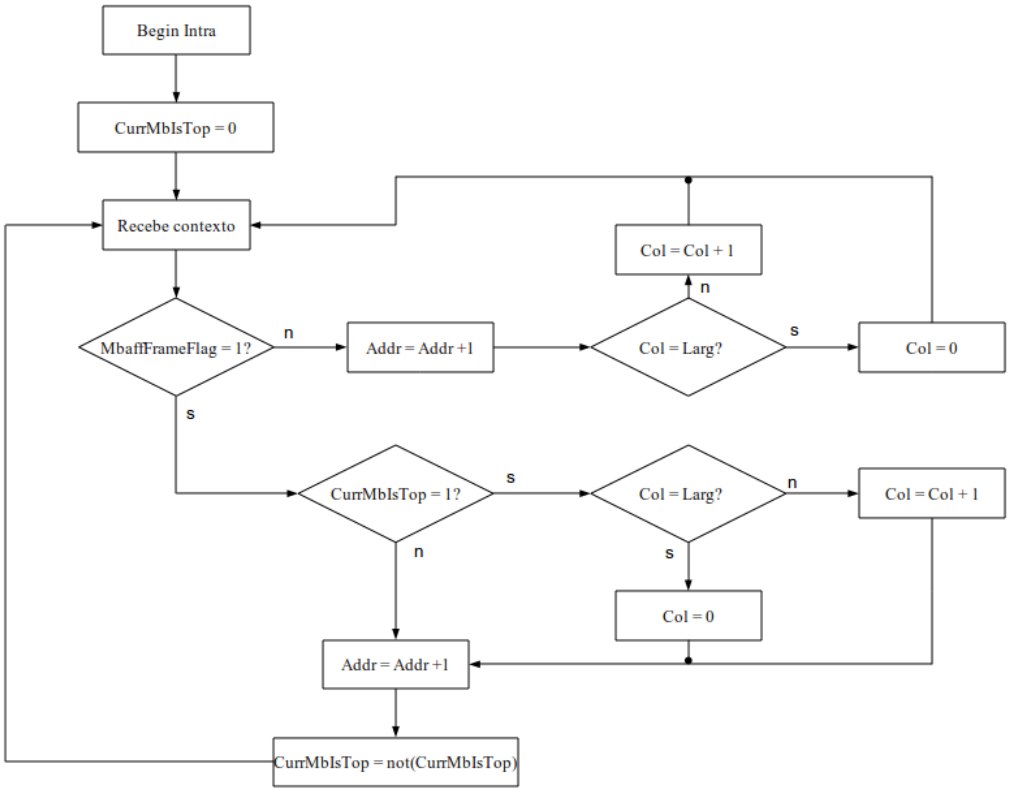

Figura 4.5: Algortimo de ajuste do *contexto*

Com isso foi criado, então, o módulo de formação de contexto denominado "*forma_ctx*".

Tendo feita esta modificação, a decodificação de *contexto*, cuja saída está ilustrada na Figura 3.2, sofre ajustes em relação a disponibilidade de vizinhos. O fluxograma presente na Figura 4.6 exemplifica a verificação de disponibilidade. A maior mudança se encontra na vizinhança superior. Dependendo do tipo de macrobloco atual, é necessário que este mesmo esteja no mínimo na terceira linha de macroblocos da imagem.

Portanto, deste modo, o contexto dos macroblocos está correto para casos de vídeo progressivo e entrelaçado.

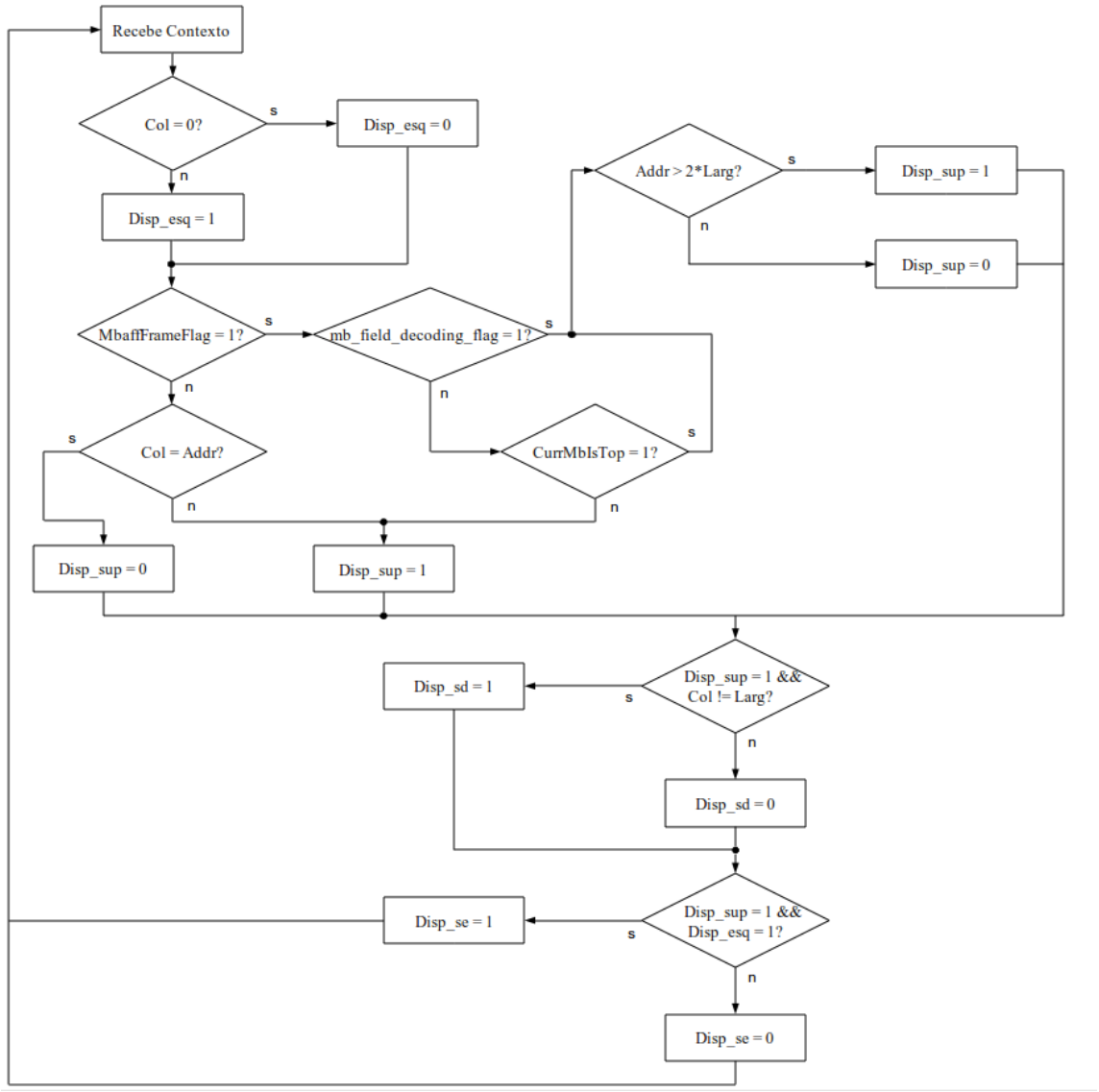

Figura 4.6: Algoritmo do cálculo de disponibilidade de vizinhanças

4.2.2 Adaptação no Armazenamento de Amostras

A presença de vídeo entrelaçado requer que sejam guardadas mais amostras que no vídeo progressivo, A partir da análise da Tabela 2.1 foi visto que os casos mais críticos requerem que sejam guardados até duas linhas de vizinhos superiores, duas colunas de vizinhos da esquerda e duas amostras de vizinhos diagonais, de acordo com a organização ilustrada na Figura 4.7.

Deste modo, em posse organização da memória de vizinhanças (*mem_viz*) a mesma foi expandida e organizada como segue a Tabela 4.1.

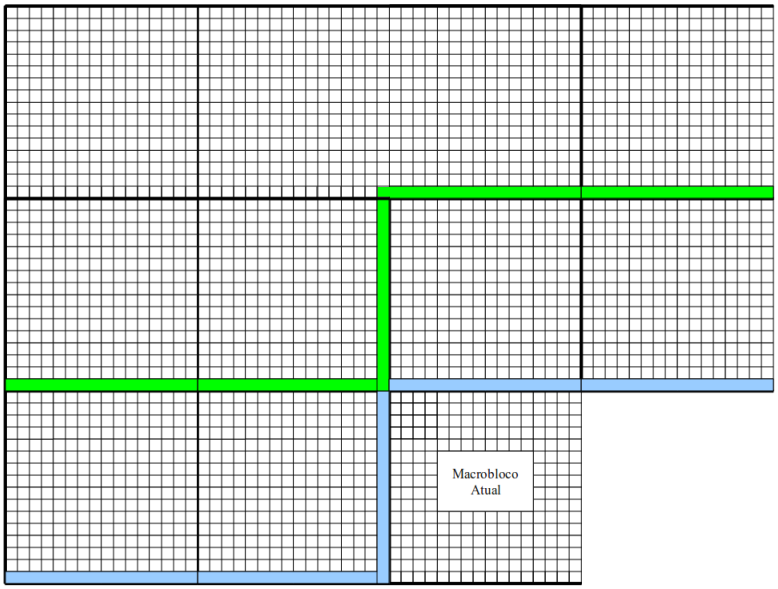

Figura 4.7: Amostras armazenadas na implementação para vídeo entrelaçado

Tabela 4.1: Nova organização da memória de vizinhanças

Posição de memória	Nº de amostras	Tipo de vizinhança
0-479	1920	Linha de vizinhos superiores de luminância do vizinho “top”.
504-507	16	Coluna de vizinhos laterais de luminância do vizinho “top”.
508-509	8	Coluna de vizinhos laterais de crominância azul do vizinho “top”.
510-511	8	Coluna de vizinhos laterais de crominância vermelha do vizinho “top”.
512-751	960	Linha de vizinhos superiores de crominância azul do vizinho “top”.
768-1007	960	Linha de vizinhos superiores de crominância vermelha do vizinho “top”.
1024-1503	1920	Linha de vizinhos superiores de luminância do vizinho “bottom”.
1528-1531	16	Coluna de vizinhos laterais de luminância do vizinho “bottom”.
1532-1533	8	Coluna de vizinhos laterais de crominância azul do vizinho “bottom”.
1534-1535	8	Coluna de vizinhos laterais de crominância vermelha do vizinho “bottom”.
1536-1775	960	Linha de vizinhos superiores de crominância azul do vizinho “bottom”.
1792-2031	960	Linha de vizinhos superiores de crominância vermelha do vizinho “bottom”.

Os novos endereços de armazenamento foram escolhidos por serem exatamente os mesmos do vídeo progressivo somados com o valor “1024” por significar o valor que um bit mais significativo de endereçamento adiciona, caso possua valor “1”. Portanto, utilizando a característica de ordem de macroblocos *top* e *bottom* presente no MBAFF, foi

utilizado justamente este indicador para guardar o dobro do número de amostras vizinhas.

Para a vizinhança diagonal, foram incluídos três registradores extras, totalizando 6, onde para cada crominância e luminancia são dedicados dois. A lógica envolvida seguiu novamente o mesmo raciocínio, selecionando qual conjunto de três registradores a partir da *flag CurrMbIsTop*.

Ajustes no armazenamento de vizinhanças 4x4 não foram realizados, pois pela norma (ITU-T, 2007) nenhuma mudança ocorre na predição em nível de bloco 4x4 quando implantado suporte a vídeo entrelaçado.

4.2.3 Modificação na Busca de Vizinhanças

Tendo a organização do armazenamento de novas amostras, descrita na seção 4.2.2, é necessário buscar corretamente os vizinhos, entretanto não se pode utilizar somente a lógica inversa do armazenamento, pois a busca dos vizinhos não segue estritamente a mesma configuração. Ao invés disso, como mencionado na seção 2.4, a busca deve passar pela análise da Tabela 2.1.

Para escolher entre os casos da Tabela 2.1 é necessário saber o *mb_field_decoding_flag* de seus vizinhos, deste modo é utilizada uma matriz de bits do tamanho da imagem em macroblocos onde cada posição armazena o valor desta *flag*. Assim é selecionada a amostra vizinha correta a partir dos indicadores:

- *mb_field_decoding_flag* do macrobloco atual
- *mb_field_decoding_flag* do macrobloco vizinho
- *CurrMbIsTop*

Nos casos 1,2,4,5,6,7,9 e 10 da Tabela 2.1 somente as *flags* acima não bastam para definir a vizinhança lateral correta. Nestes casos a vizinhança muda pixel a pixel, assim, decidiu-se manter ambas as vizinhanças laterais em registradores para poder decidir com multiplexadores qual delas realmente entra no módulo de geração de predições.

Para realizar isso foi escolhido primeiramente realizar uma busca *default* dos vizinhos usando a lógica inversa de armazenamento. Deste modo, quando o macrobloco atual é *top* os vizinhos são os "*tops*" (vide Figura 2.18) e quando o macrobloco atual é *bottom* os vizinhos são os "*bottoms*" como indicado na Figura 4.8.

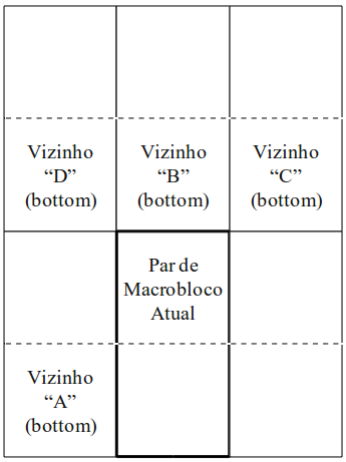

Figura 4.8: Amostras buscadas na memória quando o macrobloco atual é *bottom*

Adicionalmente para a vizinhança lateral foi criado um registrador da largura de uma coluna de amostras. Assim, de acordo com a Figura 4.9, se obtém os vizinhos adicionais em registradores. Tal algoritmo somado com a busca feita a partir das Figuras 2.18 e 4.8 faz com que sempre estejam em registradores os vizinhos *top* e *bottom*.

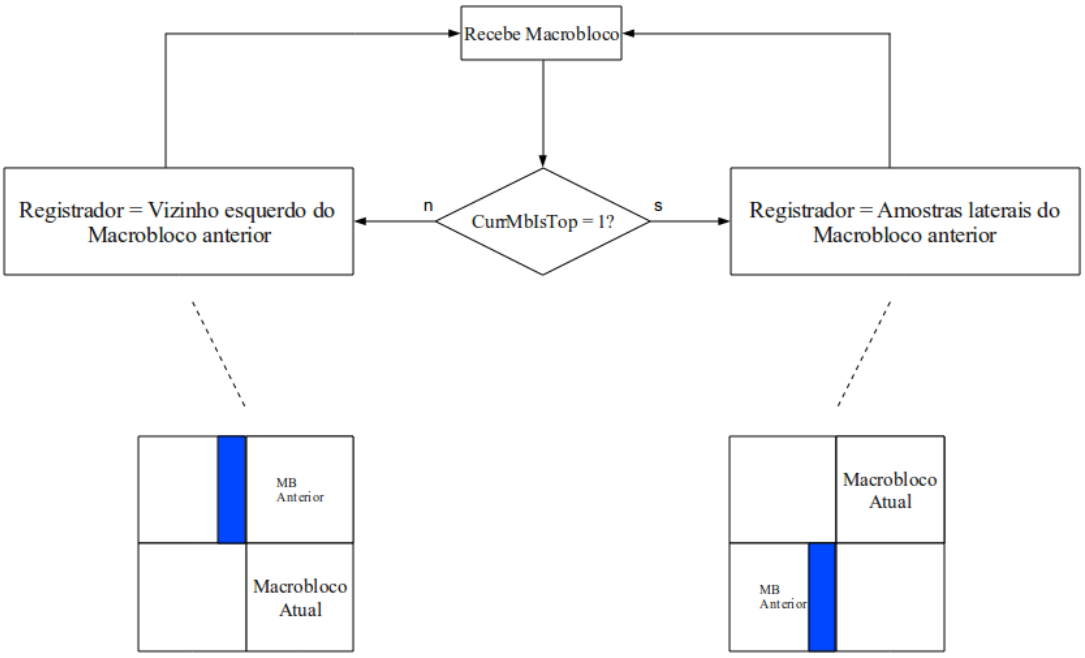

Figura 4.9: Algoritmo das amostras complementares colocadas em registrador e sua representação gráfica

Já para as vizinhanças superior e diagonal não há mudança de vizinhos no nível pixel a pixel, portanto não foi necessário o uso de um registrador auxiliar e a partir dos três indicadores já implementados é possível selecionar qual vizinho buscar na memória de vizinhança e qual registrador de amostra diagonal selecionar.

Deste modo a arquitetura resultante do Módulo de Predição Intra-quadros, bem como a comparação com a arquitetura anterior se encontra ilustrada na Figura 4.10.

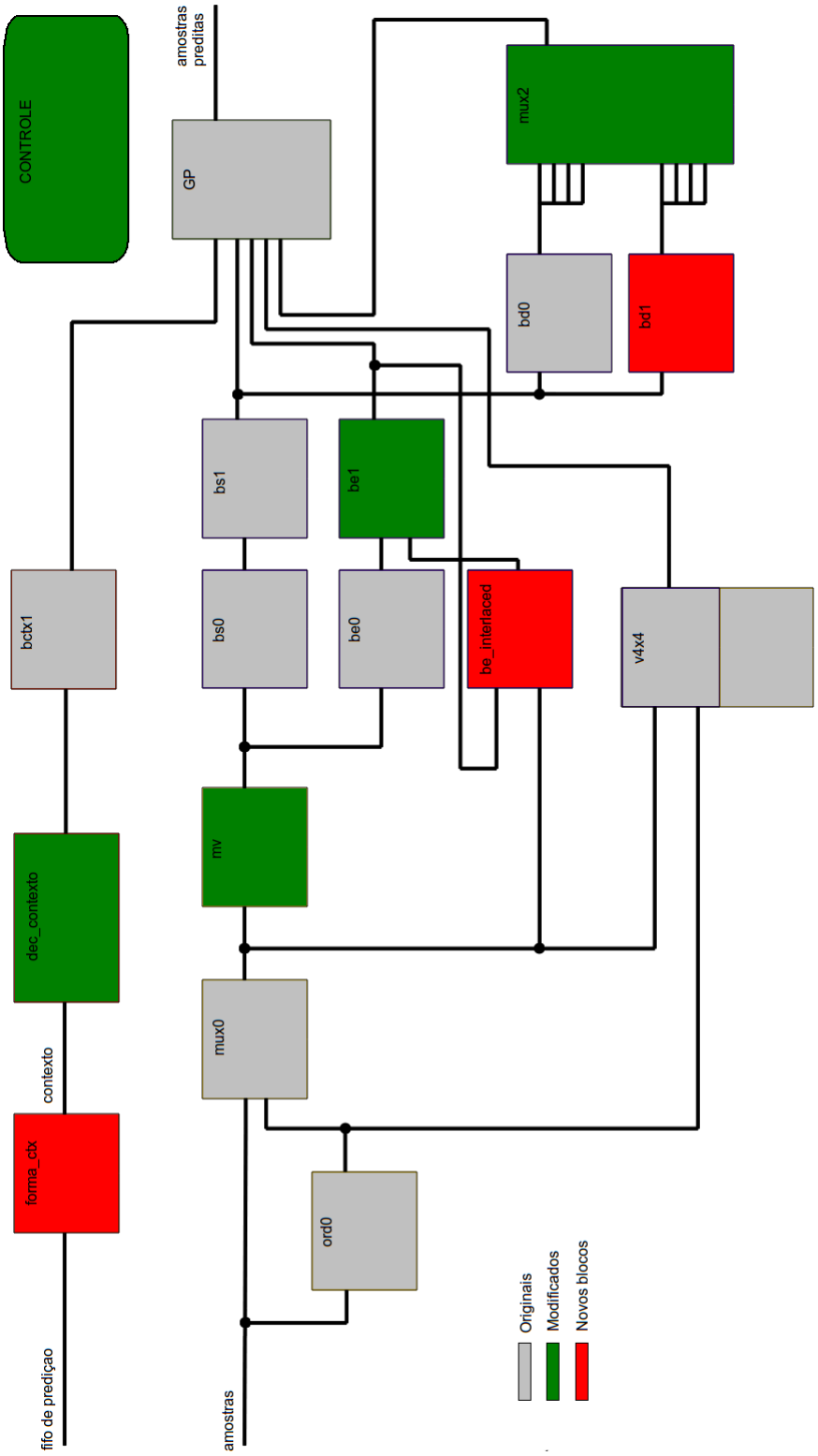

Figura 4.10: Nova arquitetura do Preditor Intra-quadros

5 RESULTADOS

Os resultados obtidos a partir da implantação das funcionalidaes descritas no capítulo 4 são divididos neste capítulo em dois níveis: resultados de validação do funcionamento e resultados de síntese para FPGA.

O primeiro descreve como o trabalho realizado teve suas funcionalidades comprovadas, as metodologias envolvidas e os programas auxiliares utilizados, bem como os resultados obtidos até o momento.

O segundo nível exibe o relatório de síntese, incluindo dados do projeto antes e após a implantação do suporte a vídeo entrelaçado.

5.1 Validação

Iniciando a parte prática do projeto, isto é, começando a realizar as primeiras modificações no módulo *parser* foi encontrado o primeiro ponto de parada em termos de teste de validação. Visto que os módulos tratados neste trabalho não fornecem uma saída absoluta do decodificador, mas sim sinais que devem passar por outros processos de manipulação para serem passíveis de teste visual, foram utilizados programas e metodologias distintos de modo a validar segmentos diferentes do trabalho.

5.1.1 Validação e Teste do Parser

O *parser* foi objeto do começo de validação, por ter sido o primeiro módulo a ser reformulado e adaptado a entrelaçamento. Para possibilitar a identificação dos erros tratados em 4.1.2 foi utilizada uma tripla comparação de valores. A saída do módulo no simulador Modelsim foi comparadas com duas referências: o analisador H.264 Visa e o Programa de Referência H.264 (PRH.264), construído na linguagem C++ pelo LaPSI.

Tal processo foi realizado, pois o analisador H.264 Visa, apesar de possibilitar acesso a muitas informações sobre o vídeo, fornece uma resposta muito abstrata, de modo que ao ocorrerem valores errados é complicada a tarefa de encontrar a fonte do problema. Já o PRH.264 tem a vantagem de estar com o código aberto, onde pode-se acompanhar os algoritmos passo a passo ou obter resultados intermediários. Entretanto podem haver erros no programa, de modo que o teste ideal é comparar a saída deste com a saída do H.264 Visa para validá-lo e utilizar suas variáveis internas para comparar até em nível "passo-a-passo" com a implementação em *hardware*.

Portanto, como o *parser* possui a funcionalidade de separar dados e interpretar o *bitstream*, a abordagem foi executar passo a passo as instruções no PRH.264 observando a quantidade de bits consumida por macrobloco e comparando com os quantidade consumida no Modelsim. Adicionalmente os valores de "nc" (vide seção 4.1.2) eram conferi-

dos.

Assim, após a reimplementação de certas partes do módulo, foi deixado este fora do escopo momentâneo, visto que a proposta inicial era implantar as funcionalidades no módulo de predição Intra-quadros.

5.1.2 Validação e Teste do Módulo de Predição Intra-quadros

A partir do conhecimento de que os dados de entrada do preditor Intra-quadros estavam corretos foi, então, implementada a proposta. Como o módulo de predição Intra-quadros oferece na sua saída amostras preditas, foi possível utilizar valores de referência recolhidos com o auxílio do H.264 Visa. Um exemplo de valores de referência e execução do H.264 pode ser visto na Figura 5.1.

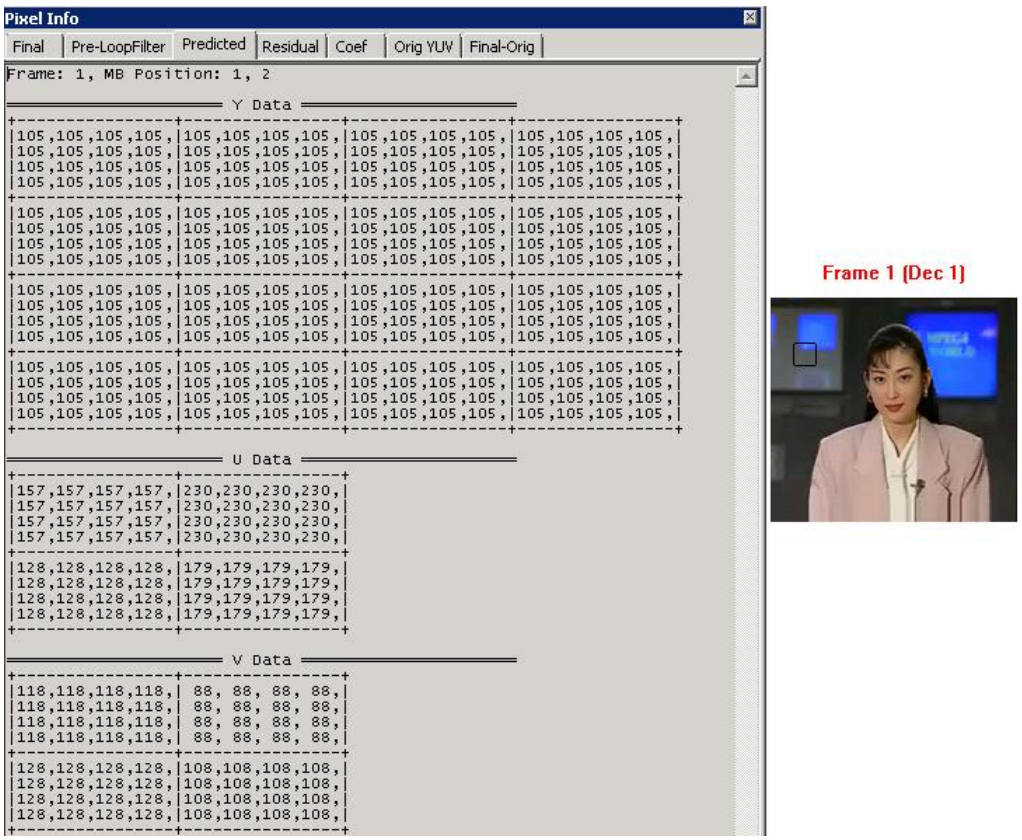

Figura 5.1: Execução do H.264 Visa com macrobloco selecionado e valores das amostras preditas atreladas

Entretanto, retirar os valores do analisador e colocá-los em um arquivo para teste é uma tarefa custosa, visto que para fins de validação são necessários muitos dados. Deste modo se resolveu utilizar o PRH.264, que gera os mesmos valores necessários, porém os coloca em arquivos texto. Portanto, a partir dos destes foram automatizados os testes, como esquematizado na Figura 5.2.

Assim, uma execução do Modelsim realizando o proposto na Figura 5.2 pode ser vista na Figura 5.3. A partir do cursor amarelo, no sinal *saida* do módulo de geração de predições, estão os valores das amostras de referência da Figura 5.1. Os sinais no topo da simulação significam o contador de bits de macrobloco acumulado até o momento, o estado de decodificação no *slice data* do *parser* e o contador de erros do módulo.

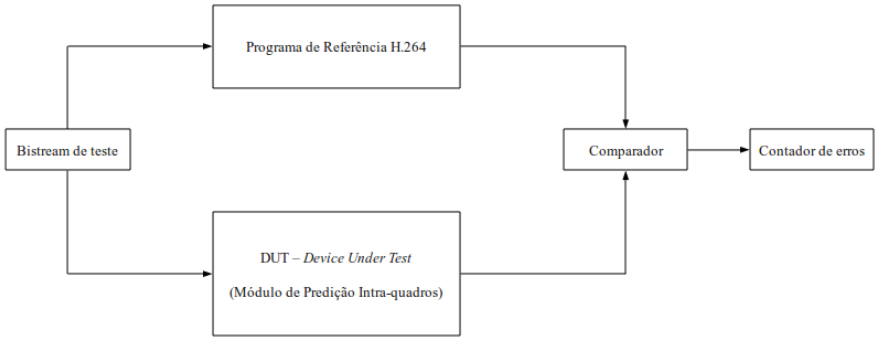

Figura 5.2: Representação do método de validação utilizado

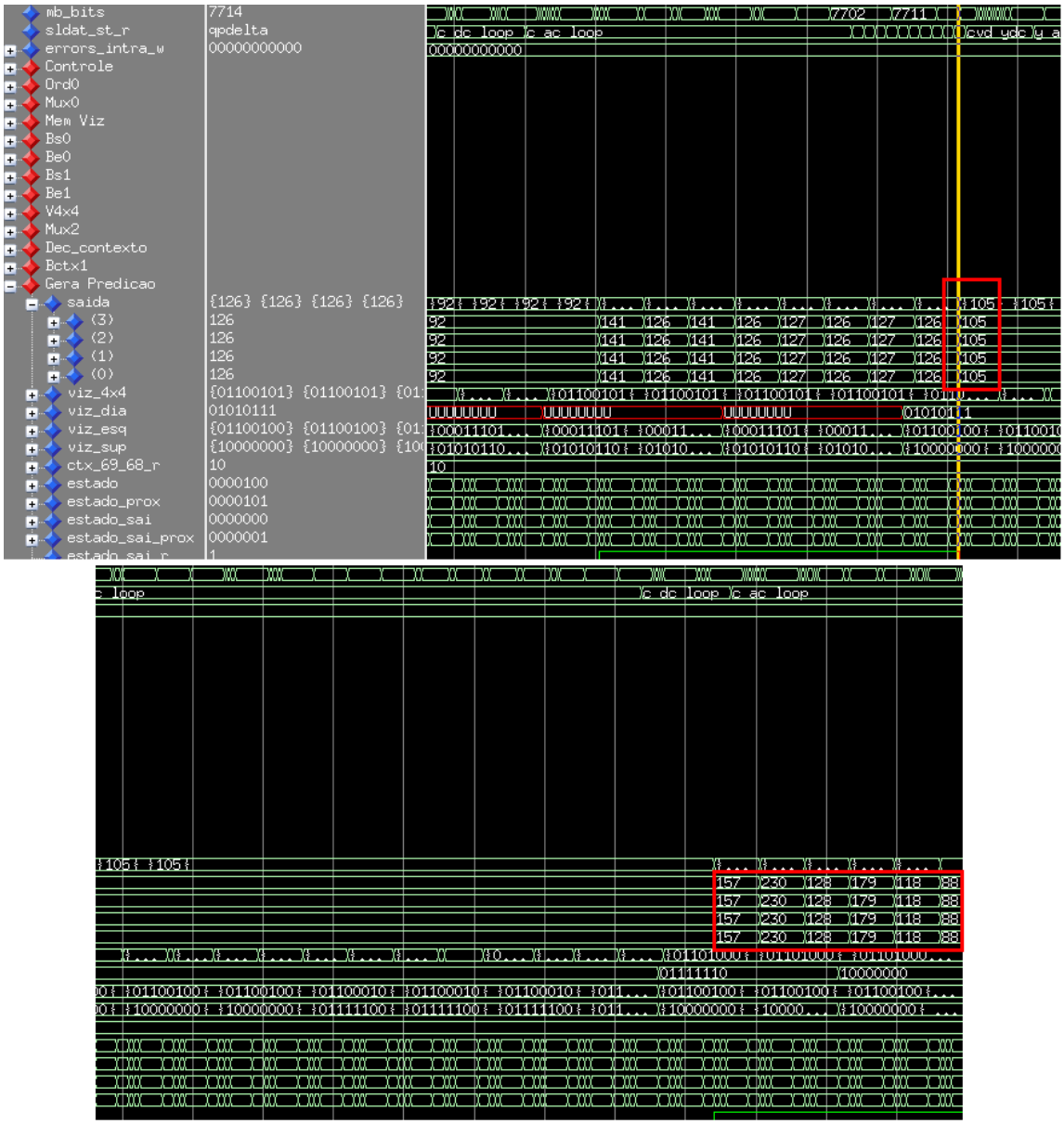

Figura 5.3: Execução do Modelsim para o modo de validação proposto na Figura 5.2

Contudo, apesar de automatizados, os testes se tornaram complexos, pois na arquitetura do preditor Intra-quadros existem dois módulos que influenciam diretamente no correto funcionamento dele. Destes módulos, o *parser*, no qual foi detectado erro nas primeiras etapas de desenvolvimento, sofreu alterações que possibilitaram a realização do projeto. Já o outro é o módulo de cálculo de resíduos *ITIQ*, que realiza as operações de transformada e quantização inversas (vide seção 2.3.3). Este não possuia suporte a entrelaçamento, sem conhecimento prévio de nenhuma pessoa envolvida no projeto.

O módulo de resíduos influencia o cálculo de predição Intra-quadros, pois ao serem geradas as predições as mesmas são somadas ao resíduo e reutilizadas pelo preditor, o que significa que o módulo de predições pode estar funcionando, mas graças a valores errados de resíduo, ele utilizaria valores de vizinhança errados e propagaria o erro para o resto do vídeo.

Para contornar este problema, então, os valores de resíduo utilizados foram provindos também do PRH.264, o que tornou o processo de validação um tanto mais demorado.

Finalmente, a validação de imagens com entrelaçamento MBAFF puderam ocorrer.

5.2 Síntese para FPGA

Tendo a validação do projeto concluída, foi então colocado em prática o processo de síntese para FPGA. Para obter o relatório de síntese, foram isolados os módulos alvos do resto do projeto e escolhido como dispositivo alvo a placa Virtex 5 XC5VLX110T. A razão para esta escolha está no fato de futuramente todo o decodificador ser integrado em um projeto e sintetizado para esta placa.

5.2.1 Síntese do *Parser*

Para o módulo *parser*, como foi reimplementado o módulo *nc* de modo a guardar o dobro de amostras em registradores e o dobro de amostras em memória, o resultado de síntese, presente na Tabela 5.1, demonstrou um aumento de área em registradores e LUTs, não havendo aumento em BRAMs pelo fato da memória extra ter sido mapeada em LUTs.

Tabela 5.1: Relatório de síntese do *parser* - Utilização de recursos

Recurso	Versão anterior (LORENCETTI, 2010)	Versão com suporte a entrelaçamento
Número de Registradores	1124 (1%)	1593(2%)
Número de LUTs	3541 (5%)	4259 (6%)
Número de BRAMs (Block Rams)	1 (0%)	1 (0%)

Os dados temporais da síntese sofreram aumento em termos de frequência não previsto, como pode ser visto na Tabela 5.2. Entretanto o aumento foi positivo, pois se diminuiu o período mínimo. Deste modo, foi possível analisar que as partes reimplementadas no *parser* continham o caminho crítico do módulo, que determina a frequência máxima.

Tabela 5.2: Relatório de síntese do *parser* - Dados temporais

	Versão anterior (LORENCETTI, 2010)	Versão com suporte a entrelaçamento
Período mínimo	7,858ns	7,397ns
Frequência máxima	127,257MHz	135.194MHz

5.2.2 Síntese do Preditor Intra-quadros

Antes de ser realizada a síntese, o resultado esperado é um aumento da área utilizada pelo preditor Intra-quadros. Como o módulo foi feito de modo a manter o funcionamento com mesmo atraso caso fosse predito vídeo progressivo e vídeo entrelaçado, sem modificar os resultados de frequência anteriores, o esperado em termos de desempenho é que não ocorram alterações significativas.

Para a síntese foi utilizado como referência o trabalho anterior de (ENÉAS, 2011), que implementou uma otimização na frequência do preditor Intra-quadros descrito em (BERRIEL, 2005).

Portanto, após executada a síntese, temos os dados de área dispostos na Tabela 5.3. Houve um aumento pequeno no número de registradores (0,06%), diminuição no número de LUTs (0,02%), devido provavelmente a otimizações de trechos do código em VHDL, e um aumento no número de Block Rams de “2” para “3”. Neste caso o aumento mais significativo foi de BRAMs.

Tabela 5.3: Relatório de síntese do Preditor Intra-quadros - Utilização de recursos

Recurso	Versão anterior (ENÉAS, 2011)	Versão com suporte a entrelaçamento
Número de Registradores	2110 (3,05%)	2149 (3,11%)
Número de LUTs	4428 (6,41%)	4414 (6,39%)
Número de BRAMs (Block Rams)	2 (1,35%)	3 (2,03%)

Os dados temporais estão dispostos na Tabela 5.4. O pequeno aumento de frequência provavelmente é provindo da reorganização dos blocos no circuito, que alteraram levemente o caminho mais longo que determina o período mínimo. Deste modo, desconsiderando este fato, é possível se observar que a frequência não possui alteração e então as previsões de síntese se provaram corretas.

Tabela 5.4: Relatório de síntese do Preditor Intra-quadros - Dados temporais

	Versão anterior (ENÉAS, 2011)	Versão com suporte a entrelaçamento
Período mínimo	7,568ns	7,514ns
Frequência máxima	132,129MHz	133,091MHz

A partir da síntese de ambos os módulos alvos foram obtidos resultados satisfatórios, pois eram esperados aumentos em área. Apesar do aumento ter sido significante em comparação com os módulos de referência, ainda assim foram pequenos em relação ao número total de recursos disponíveis na placa. Do ponto de vista para integração com o resto do projeto, portanto, não há implicações críticas em termos de área.

Um dos objetivos secundários seria manter a frequência de operação dos módulos intacta, de modo a priorizar o desempenho, que foi tema de (ENÉAS, 2011). Portanto, os resultados de síntese em termos de tempo foram satisfatórios também, por não diminuir a frequência máxima de operação.

6 CONCLUSÕES

Este trabalho apresentou as etapas de estudo, análise e desenvolvimento do suporte a vídeo entrelaçado no módulo de predição Intra-quadros. Foi obtido o conhecimento necessário sobre o conceito de vídeo entrelaçado e o funcionamento da predição Intra-quadros de modo a realizar o proposto inicialmente. Adicionalmente se realizam alterações não previstas no *parser* de modo a possibilitar a implementação do preditor.

A arquitetura utilizada prezou por deixar o funcionamento para vídeo progressivo inalterado, chaveando funcionalidades novas do vídeo entrelaçado a partir de *flags*, sempre procurando realizar o mínimo de alterações nos formatos de entrada e saída de cada módulo, tendo em mente os outros módulos do decodificador que interagem com o preditor Intra-quadros.

A re-engenharia de certos módulos aumentou a área utilizada, entretanto tal resultado era esperado, visto que a complexidade é muito maior ao lidar com entrelaçamento. Ao analisar os incrementos nos recursos utilizados, entretanto, foi observado que em termos de porcentagem o aumento é pequeno em detrimento da nova funcionalidade.

A partir da realização de um projeto grande, foi possível observar a importância de metodologias para resolver um problema. Levantar requisitos, analisar possibilidades, criar estratégias para atacar o problema e construir algoritmos são vitais a medida que a complexidade do projeto em questão aumenta.

Finalmente, o trabalho se provou desafiador e satisfatório, pois não somente se obteve conhecimento sobre vídeo digital, mas também sobre as ferramentas, linguagem e conceitos de um projeto de descrição de *hardware*.

Para trabalhos futuros estão testes em placa do módulo, visando o fato que mesmo com simulações perfeitas ainda não é possível prever possíveis erros de placa, que podem acontecer ao programar a FPGA.

REFERÊNCIAS

ABNT. NBR15602. **ABNT. NBR15602 Televisão Digital Terrestre:** Codificação de vídeo, áudio e multiplexaçãoo. 2007.

BERRIEL, E. A. **O Padrão de Codificação de Vídeo H.264 e uma Implementação do Módulo de Predição Intra Quadro em *Hardware***. 2005. 71 f. Trabalho Individual (Mestrado em Ciência de Computação) - Instituto de Informática, UFRGS, Porto Alegre.

DIÁRIO OFICIAL DA UNIÃO. **Atos do Poder Executivo**. Decreto No 4.901. Número 231 de 27 de nov. de 2003. Publicado no Diário Oficial da União.

ENÉAS, D. **Otimização do Módulo de Predição Intra-Quadros e Integração em um Decodificador de Vídeo para o SBTVD**. 2011. 43 f. Projeto de Diplomação (Bacharelado em Engenharia Elétrica) - Departamento de Engenharia Elétrica, UFRGS, Porto Alegre.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation H.264. Advanced video coding for generic audiovisual services:** Audiovisual and Multimedia Systems. Infrastructure of audiovisual services - Coding of moving video. 2007.

KAMACI, N.; ALTUNBASAK, Y. Performance Comparison of the Emerging H.264 Video Coding Standard with the Existing Standards. **ICME '03 Proceedings of the 2003 International Conference on Multimedia and Expo - Volume 2**. Georgia Institute of Technology - Atlanta, Pages 345-348, 2003.

LORENCETTI, M. A. **Parser em VHDL para decodificador de vídeo H.264 para SBTVD**. 2010. 51 f. Projeto de Diplomação (Bacharelado em Engenharia Elétrica) - Departamento de Engenharia Elétrica, UFRGS, Porto Alegre.

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

SCHMIDT, A. A. A. **Integração do Cabad ao Decodificador de Vídeo H.264/AVC para o SBTVD**. 2011. 88 f. Projeto de Diplomação (Bacharelado em Engenharia Elétrica) - Departamento de Engenharia Elétrica, UFRGS, Porto Alegre.

ANEXO <ARTIGO TG1>

Implantação do Suporte a Vídeo Entrelaçado ao Módulo de Predição Intra-quadros para o SBTVD

Henrique Awoyama Klein[1], André Borin Soares[2](Co-orientador), Altamiro Amadeu Susin[2](Orientador)

[1]Instituto de Informática
[2]Departamento de Engenharia Elétrica
Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre, RS – Brasil
haklein@inf.ufrgs.br,{andre.borin, altamiro.susin}@ufrgs.br

Abstract. *This article is situated within the context of the H.264/MPEG-4 standard project, aiming more specificaly on developing the Intra-prediction module suporting interlaced video starting from a minimal video decoder previously implemented.*

Resumo. *Este artigo se situa no contexto do projeto do decodificador de vídeo para o padrão H.264/MPEG-4, visando mais especificamente no desenvolvimento do módulo de predição Intra-quadros com suporte a entrelaçamento de vídeo a partir de um decodificador de vídeo minimal previamente implementado.*

1. Introdução

1.1. Contexto

A Universidade Federal do Rio Grande do Sul, em parceria com outros centros de pesquisa acadêmicos do país, está em cargo do desenvolvimento do decodificador de vídeo no padrão H.264/MPEG-4 para o Sistema Brasileito de Televisão Digital (SBTVD), cuja inicialização se deu a partir do decreto Nº 4.901 em 26 de Novembro de 2003 [da Silva 2003]. Um dos centros de pesquisa encarregado do desenvolvimento do projeto é o Laboratório de Processamento de Sinais e Imagens (LaPSI), situado no Departamento de Engenharia Elétrica (DELET) da UFRGS. Neste ambiente está sendo desenvolvido o projeto de um decodificador em Hardware para decodificação de vídeo em tempo real, que permitirá o futuro projeto de um chip para tal processo.

Dentre as motivações para a realização do projeto está a necessidade do país de possuir um decodificador construído em território nacional, de forma a dominar a tecnologia de implementação do SBTVD futuramente diminuindo o custo, assim beneficiando a maior parte da população brasileira.

No padrão H.264/MPEG-4 existe uma vasta quantidade de recursos de compressão de vídeo utilizados sendo um deles o módulo de predição Intra-quadros, que visa explorar a redundância espacial em um mesmo quadro realizando predição a partir de partes já decodificadas do mesmo. O projeto atualmente possui esse módulo de predição em um perfil *baseline*, isto é, somente realizando predições para vídeos sem entrelaçamento.

1.2. Objetivos

Com base no contexto atual do projeto é visado nesse trabalho extender as funcionalidades do módulo de predição Intra-quadros de modo a suportar a decodificação de qualquer tipo de vídeo.

Como visto em [ITU 2007] as imagens recebidas pelo decodificador podem assumir 3 tipos do ponto de vista de predição: Imagem tipo *frame*, imagem tipo *field* e imagem tipo *MBAFF* (*Macroblock Adaptative Frame Field*); sendo a primeira característica de vídeo progressivo e as últimas 2 de vídeo entrelaçado. Os tipos de imagens serão explicadas mais extensivamente nos próximos itens desse artigo.

Portanto atualmente temos que o sistema segue a Tabela 1. De modo que as funcionalidades ainda não atendidas são o objetivo final do trabalho.

Tabela 1. Funcionalidades do módulo

Tipo de Imagem	Video entrelaçado	Atualmente implementado
Imagem *frame*		X
Imagem *field*	X	
Imagem *MBAFF*	X	

1.3. Resultados Esperados

No fim do trabalho é esperado que o módulo consiga realizar predição Intra-quadros em qualquer tipo de imagem, assim atendendo aos pré-requisitos de um decodificador de vídeo para o padrão H.264.

2. Base Teórica

2.1. Padrão H.264

A motivação do padrão H.264 vem da necessidade de comprimir drasticamente os dados de modo a possibilitar transmissão. Tal fato se demonstra quando calculadas as taxas de transmissão requeridas para fornecer vídeo sem nenhuma compressão. Tendo uma banda de 6 MHz por canal e se otimizando técnicas de modulação é possível atingir taxas de aproximadamente 20 Mbits/segundo em um canal, entretando para transmitir um vídeo em HDTV sem compressão as taxas deveriam ser na ordem de 1 Gbit/segundo.

Outro problema é o armazenamento desses dados, visto que para um vídeo SDTV de duas horas seriam gastos cerca de 240 Gbytes de memória, ao passo que o mesmo vídeo comprimido teria o gasto de memória reduzido para 1 Gbyte. Portanto o padrão trabalha com modos de compressão que exploram redundâncias inerentes ao vídeo (temporal e espacial) e ao ser humano (de frequência).

2.2. Arquitetura do Decodificador

A arquitetura de um decodificador de vídeo H.264/MPEG-4 segue a Figura 1. Os dados passam por uma operação de decodificação de entropia, seguidos de descompressão de elementos sintáticos e parâmetros, recuperação dos resíduos e realização de predições. A imagem recuperada da soma da predição com o resíduo passa então por um processo

de filtragem. Dentre os módulos Inter e Intra é escolhido o primeiro caso se explore redundância temporal, ou o segundo caso se explore redundância espacial.

O funcionamento detalhado de cada bloco será omitido, visto que o foco do trabalho se dará no módulo de Predição Intra-quadros, cujo funcionamento será explicado na sessão seguinte. Para maior detalhamento da arquitetura vide [Richardson 2003].

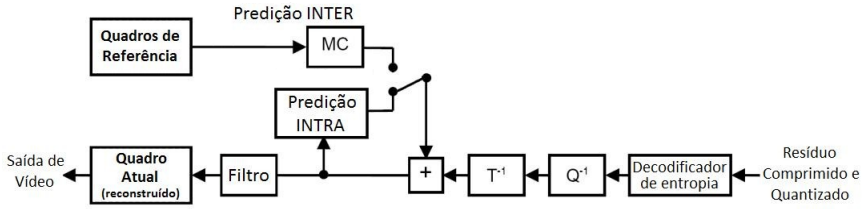

Figura 1. Arquitetura do Decodificador H.264/MPEG-4

2.2.1. Exploração de Redundâncias

As redundâncias exploradas pelo padrão H.264 são categorizadas em: temporal, espacial e de frequência. A primeira se caracteriza pela semelhança entre imagens em tempos distintos, visto que muitas vezes de uma imagem a outra existem poucas variações, ocorrendo movimentos em fragmentos da mesma, esses indicados por *vetores de movimento* como podemos ver na Figura 2. A segunda trabalha em identificar as redundâncias entre regiões homogêneas da mesma imagem, visto na Figura 3. Finalmente a terceira parte do pressuposto que o olho humano possui maior sensibilidade a baixas frequências, podendo-se assim se desprezar variações pequenas em sinais em altas frequências.

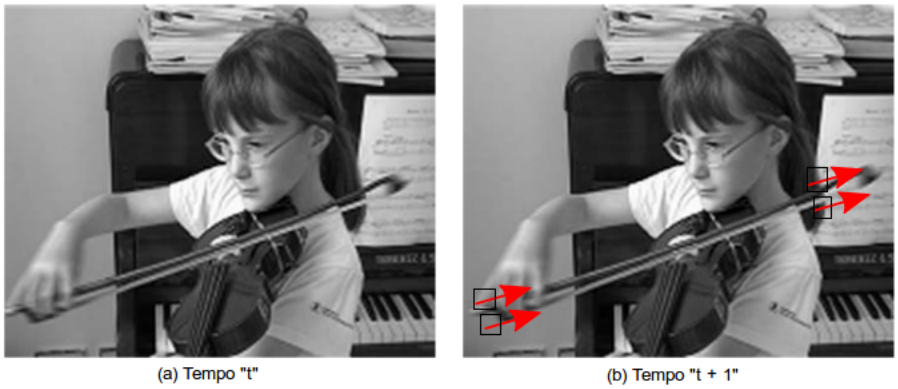

Figura 2. Redundância temporal e indicação do sentido do movimento

Para cada tipo de redundância existe um módulo que a explora. Em termos de redundância temporal tem-se o Preditor Inter-quadros, que determina os vetores de movimento, ao passo que para espacial temos o Preditor Intra-quadros, que será explicado na sessão 2.3, e enfim o Módulo de Quantização para eliminação de frequências desnecessárias.

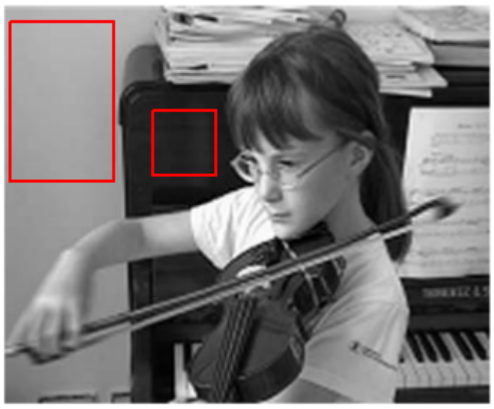

Figura 3. Redundância espacial e indicação de regiões homogêneas

2.2.2. Espaços e Sub-amostragem de Cor

Na representação de imagens, qualquer cor pode ser decomposta com uma soma das componentes vermelha(R), verde (G) e azul (B) formando então o espaço RGB. Entretando ao estudar o olho humano se percebe que o mesmo possui mais sensibilidade a luz que a cores. É possível calcular a componente de luminância (Y) a partir de uma soma ponderada das componentes RGB e então subtrai-la das mesmas, gerando as denominadas crominâncias (Cr, Cg *e* Cb).

Contudo somente são necessárias duas das três crominâncias, visto que a partir da informação de luminância e duas crominâncias é possível se deduzir a terceira. Assim, são informadas a luminância (Y), crominância azul (Cb) e crominância vermelha (Cr) formando o espaço $YCbCr$ que possibilita ter resoluções diferentes para luz e cor, ao passo que em RGB todas as componentes são normalmente representadas com a mesma resolução.

Portanto tomando posse desse formato se realiza sub-amostragem de cores, onde se escolhem proporções para as quais serão impostas as componentes de luz e cor. Normalmente se utilizam três padrões: 4:4:4, 4:2:2 ou 4:2:0. Essas relações definem quantas amostras de crominância existem para amostras de luminância. A Figura 4 ilustra as sub-amostragens mencionadas.

2.2.3. Macroblocos e Blocos

No padrão H.264 uma imagem é dividido em *macroblocos* de 16x16 pixels e no caso de uma sub-amostragem de 4:2:0 existem 16x16 amostras de luminância e 8x8 para cada crominância. Por sua vez cada macrobloco é dividido em blocos de 4x4 pixels e portanto 4x4 amostras de Y, 2x2 amostras tanto de Cb quanto de Cr. As Figuras 5 e 6 mostram, respectivamente, uma imagem QCIF (176x144 pixels) dividido em macroblocos e blocos de luminância e crominância, bem como a ordem de varredura de ambos.

Ao ser decodificado video entrelaçado será visto na sessão 4 que a ordem de varredura dos blocos é modificada.

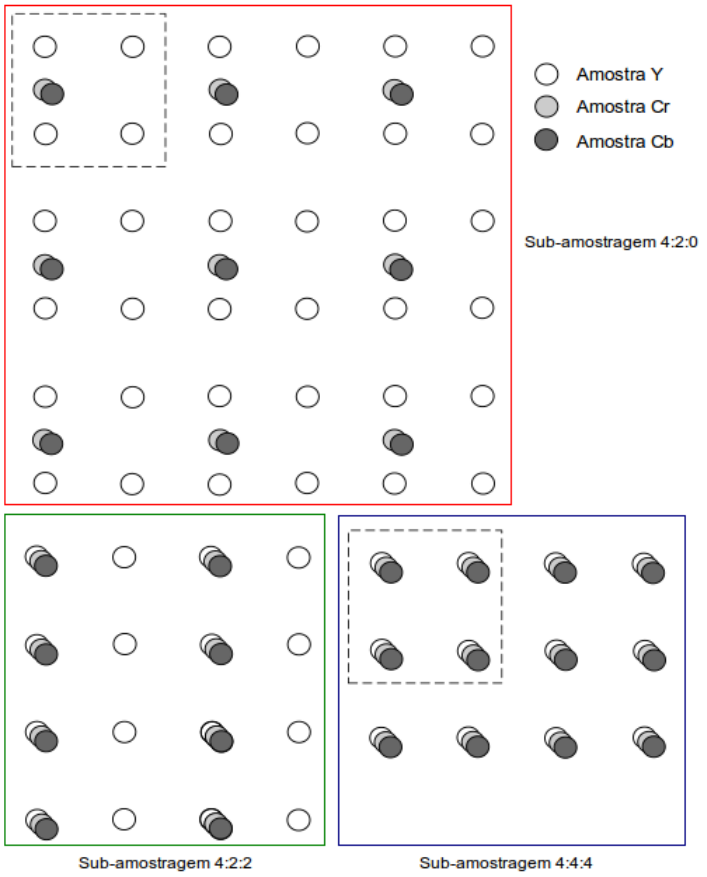

Figura 4. Sub-amostragens de cores

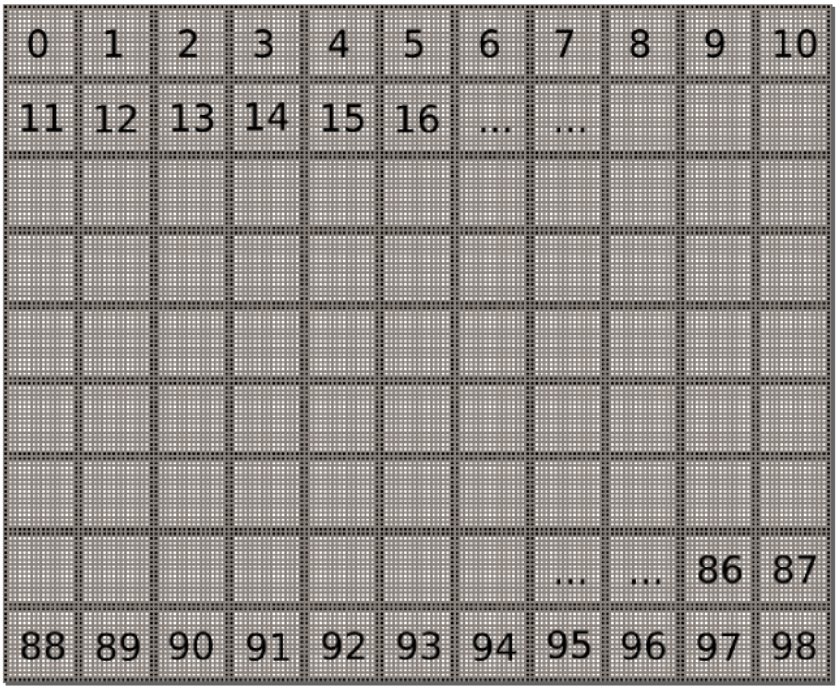

Figura 5. Ordem de varredura de macroblocos em uma imagem QCIF (176x144)

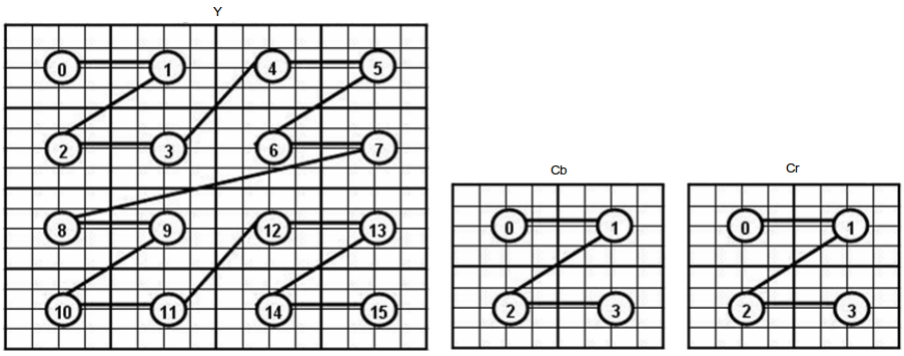

Figura 6. Ordem de varredura de amostras de luminância e crominâncias

2.3. Predição Intra-quadros

No módulo de predição Intra-quadros não há necessidade de se manterem imagens de referência inteiras, pois a predição é feita a partir de partes da própria imagem , mais próximas ao bloco atual, que já foram decodificadas .As partes que servem como parâmetros para determinar o valor de predição do bloco são denominadas *vizinhanças* e estão dispostas a esquerda e acima do bloco a ser predito, como visto na Figura 7. O cálculo do valor predito do pixel se dá por meio de uma cópia ponderada dos valores dos vizinhos do bloco.

M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

Figura 7. Bloco de pixels a serem preditos (em minúsculo) e vizinhanças (em maiúsculo)

Existem modos de predição em dois níveis em termos de granularidade: predição em blocos 4x4 ou blocos 16x16. A Figura 8 evidencia os 9 modos de predição 4x4 de luminância que indicam quais vizinhos serão utilizados no cálculo da predição.

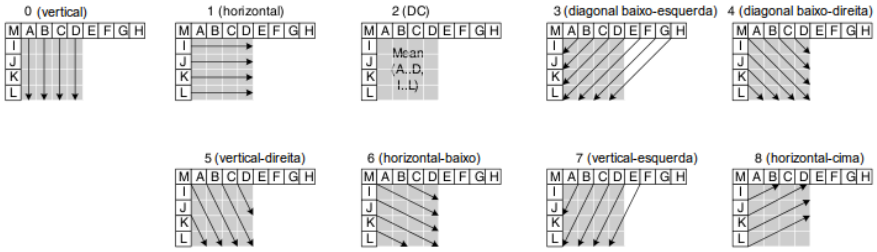

Figura 8. Modos de predição Intra-quadros 4x4

Os modos de predição 16x16 de luminância estão demonstrados na Figura 9. Esses mesmos modos são utilizados para predição de crominâncias, exceto que nesses casos o modo *DC* é o "0" e o modo *vertical* é o modo 2.

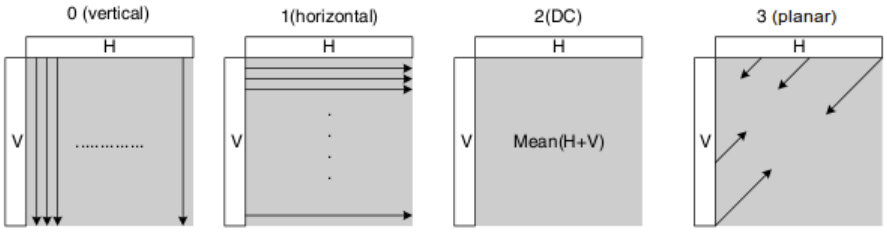

Figura 9. Modos de predição Intra-quadros 16x16

2.4. Vídeo Entrelaçado

O vídeo entrelaçado se caracteriza por introduzir o conceito de macrobloco tipo *field* ou "campo", cuja funcionalidade é a divisão entre *top field* e *bottom field* de um par de macroblocos. Este tipo de codificação é baseado no formato de vídeo de TV analógica, no qual as linhas pares e ímpares são varridas (e atualizadas) em instantes diferentes de tempo. No padrão H.264 este conceito pode ser aplicado tanto a imagens inteiras quando pares de macroblocos . Deste modo, os macroblocos, organizados em pares *top* e *bottom*, e são complementares entre si. A Figura 10 ilustra um par de macroblocos tipo *field*.

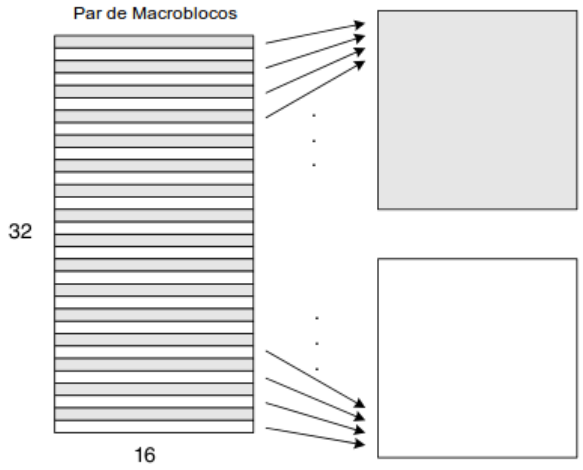

Figura 10. Par de Macroblocos tipo *field*

Assim o tipo da imagem sendo *field* (também chamado de PAFF - *Picture-Adaptative Frame Field*) ou MBAFF (*Macroblock-Adaptative Frame Field*) acarreta a utilização de macroblocos tipo *field* e portanto implica em mudanças na ordem de decodificação e principalmente nas vizinhanças do macrobloco. Tais mudanças serão tratadas na sessão 4.

3. Trabalhos Relacionados

3.1. Módulo de Predição Intra-quadros

Esse trabalho foi desenvolvido pelo engenheiro Eduardo Agostini Berriel no Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande do Sul [Berriel 2005] e trata do projeto e implementação do Módulo de Predição Intra-quadros

em VHDL em um perfil *baseline*, ou seja, realizando predições para vídeos progressivos (imagens tipo *frame*).

O algoritmo utilizado em [Berriel 2005], presente na Figura 11 realiza primeiramente a detecção do tipo de predição: 4x4 ou 16x16, seguindo por verificar a disponibilidade da vizinhança (atento ao fato que macroblocos pertencentes a primeira linha não possuem vizinhos superiores e macroblocos pertencentes a primeira coluna não possuem vizinhos a esquerda) e caso existam vizinhos disponíveis e esses estejam com permissão para predição intra-quadros os modos de predição são derivados.

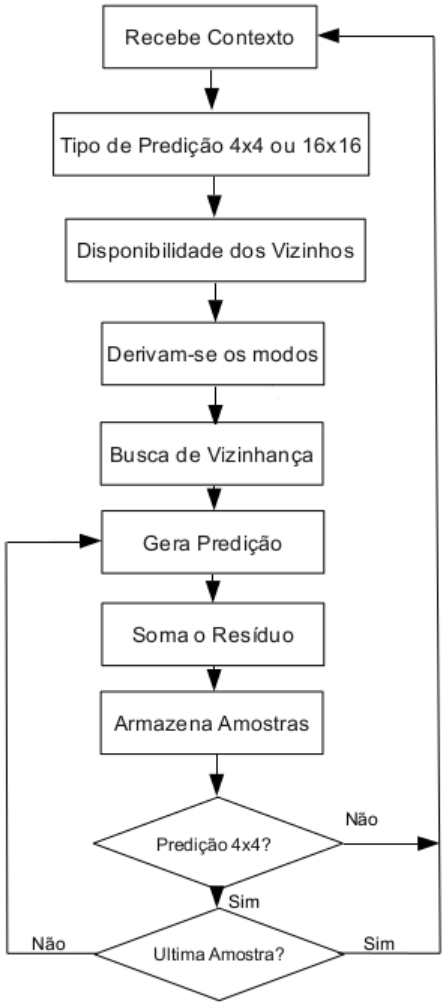

Figura 11. Algoritmo de Predição Intra-quadros

Para implementar todo o procedimento descrito existem quatro divisões funcionais: Decodificação de Contexto, Busca de Vizinhanças, Geração de Predições e Armazenamento de Amostras. As funções atreladas dizem respeito a decodificar o contexto do macrobloco atual (modo de predição, resolução, posição do mesmo em relação ao quadro, etc) e a partir dessas informações buscar os seus vizinhos podendo então gerar a predição e finalmente armazenar as amostras preditas para servirem como vizinhos a futuros blocos/macroblocos.

A partir desse módulo foi implementado em [Enéas 2011] a integração do mesmo

com o decodificador de vídeo e aumento da frequência de operação pelo engenheiro Dierles Enéas. Com base nesses trabalhos que será implementado a proposta desse artigo.

4. Arquitetura Proposta

A partir da subdivisão funcional realizada no item 3.1 será proposto implementar suporte a vídeo entrelaçado no Módulo de Predição Intra-quadros *baseline* implementado em [Berriel 2005]. De acordo com a sessão 2.4 deverão ser realizadas mudanças na ordem de varredura dos macroblocos e alteração nas vizinhanças dependendo do modo de predição.

Entretanto, para realizar as implementações é necessário primeiramente identificar os dados que sinalizam qual tipo de vídeo está sendo decodificado: progressivo ou entrelaçado. Assim foi realizado o estudo do *parser* de decodificação do padrão H.264 [Lorencetti 2010] e propagadas para a entrada do Preditor Intra-quadros as *flags* indicadoras do tipo de imagem:

- *MbaffFrameFlag*: Indica se a imagem é *Frame-Field* Adaptativo, ou seja, se na mesma existem tipos diferentes de macroblocos;
- *mb_field_decoding_flag*: Indica se o macrobloco atual é *field* ou *frame*.

Portanto analisando os sinais e relacionando-os com os tipos de imagens temos a Tabela 2.

Tabela 2. Relação entre *flags* e tipos de vídeo

MbaffFrameFlag	*mb_field_decoding_flag*	Imagem	Macrobloco	Vídeo
0	0	*frame*	*frame*	progressivo
0	1	*field*	*field*	entrelaçado
1	0	*MBAFF*	*frame*	entrelaçado
1	1	*MBAFF*	*field*	entrelaçado

A partir da análise da Tabela 2 é possível então observar que 3/4 dos casos não foram implementados e pela complexidade intrínseca ao entrelaçamento devem acarretar mudanças muito significativas nos módulos do Preditor.

4.1. Requisitos Funcionais

Para implementar alterações no Preditor Intra-quadros primeramente são descritas os requisitos funcionais independente de implementação, e então elencados os módulos os quais essas novas funcionalidades impõem reformulação.

4.1.1. Ordem de Decodificação

A ordem de decodificação descrita na sessão 2.2.3 nas Figuras 5 e 6 é modificada quando se possui entrelaçamento, visto que os macroblocos são organizados em pares como ilustrado na Figura 12. Entretando, no contexto do macrobloco a varredura das amostras no Módulo Intra ainda são as mesmas evidenciadas na Figura 6.

Portanto seguindo a ordem mostrada na Figura 12 será necessário identificar que o macrobloco vem em pares *top* e *bottom* e assim mudar a atualização da posição do macrobloco atual em linha e coluna.

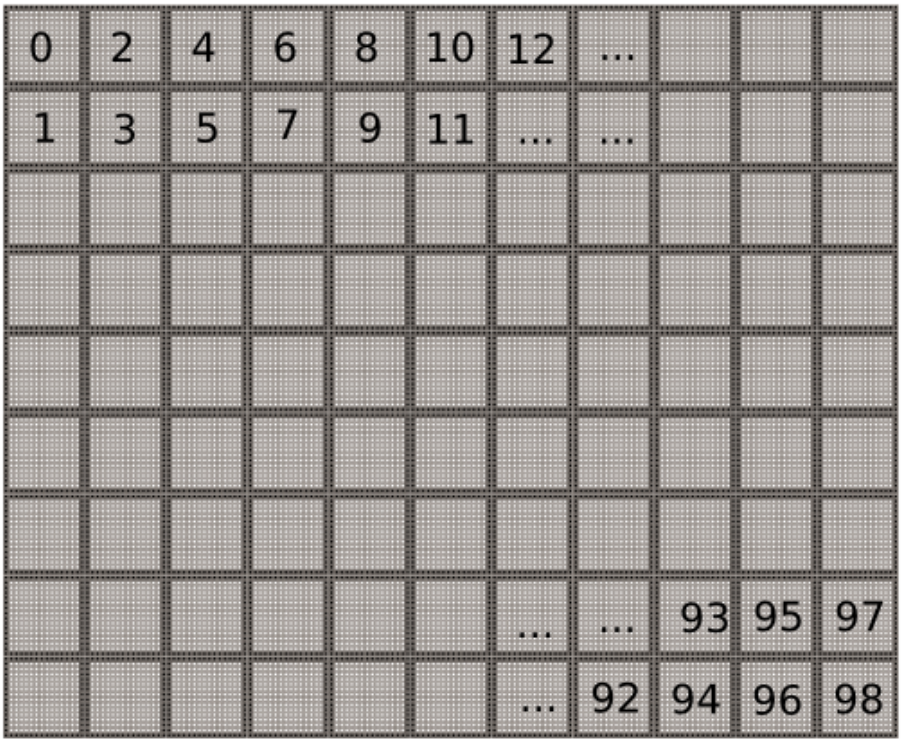

Figura 12. Ordem de varredura de macroblocos em uma imagem QCIF (176x144) com entrelaçamento

4.1.2. Busca de Vizinhanças

De acordo com a Norma [ITU 2007] as amostras vizinhas requisitadas dentro de um macrobloco (predição 4x4) não sofrem alteração, entretanto, como a ordem de decodificação é modificada (vide Figura 12) os modos de predição 16x16 tem a sua vizinhança alterada, e dada a variação entre pares *frame* e *field* existem múltiplas combinações possíveis. Tomando como base a Figura 13 pode-se ter os macroblocos A, B, C, D e Atual variando entre *frame* e *field* (caso imagem MBAFF), fazendo que o processo de obter vizinhanças relacionando-se tipos diferentes de macroblocos seja bem mais complexo que o realizar somente para tipos *frame*.

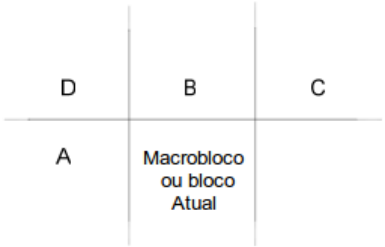

Figura 13. Representação das vizinhanças de um bloco/macrobloco

4.2. Módulos a Serem Modificados

A partir da análise das funcionalidades que devem ser implementadas é possível estruturar no contexto do projeto em Hardware os novos módulos a serem integrados.

4.2.1. Decodificador de Contexto

O decodificador de contexto atual decodifica a estrutura de contexto de um macrobloco, que contém todas as informações necessárias para processá-lo:

- Posição relativa a imagem;
- Posição relativa à linha;
- Largura da imagem em nº de macroblocos;
- Tipo de Sub-amostragem;
- Tipo de Predição de luminância;
- *Flags* de quais blocos 4x4 tem modo derivado por predição;
- Vetor com modos entre 0 e 7 dos blocos 4x4 que não tem o seu modo derivado por predição;
- Modo de predição de crominâncias;
- *Flag* indicando se o macrobloco pode ser usado como vizinho;

Também existe na estrutura do contexto um espaço dedicado ao *flag* indicando o tipo de imagem (progressivo, PAFF ou MBAFF), porém não chegou a ser implementado. Portanto implementação de entrelaçamento no decodificador de contexto diz respeito a incluir a informação do tipo de imagem a fim de tomar decisões de controle para outros módulos.

4.2.2. Memória de Vizinhanças

A memória de vizinhanças armazena somente os vizinhos necessários para a predição do macrobloco atual em modo progressivo de imagem, porém a partir do visto na sessão 4.1.2 a memória deverá ser reformulada de modo a guardar mais blocos, visto que talvez seja necessário ter como referência um macrobloco *top* previamente predito, por exemplo, que em um vídeo progressivo não seria mais guardado em memória.

4.2.3. Controle

Deverá ser descrito um Controle novo para o Preditor, pois nele está a função de chavear as memórias de vizinhança e fornecer endereços de leitura e escrita, onde ocorrem as maiores alterações. Assim a responsabilidade de atender a ordem de decodificação dos macroblocos (Figura 12) e buscar as vizinhanças corretamente depende da implementação do módulo.

5. Atividades Futuras

Com as atividades descritas nas sessões anteriores o estudo do módulo foi realizado de tal maneira que será possível realizar as funcionalidades exigidas para o desempenho total do Módulo de Predição Intra-quadros.

5.1. Implantação das Modificações

Primeiramente será de fato descrito em Hardware os módulos mencionadas na sessão 4.2 e quaisquer mais modificações adicionais que forem decorrente dessas.

5.2. Validação do Projeto

Após o término da descrição é necessário validar o seu resultado em conjunto com o decodificador de vídeo. Para isso serão utilizados as estratégias:

- Comparação de resultados com o Programa de Referência H.264 , cuja funcionalidade é executar em Software o mesmo processo de decodificação a fim de comparar os resultados e validar o projeto em Hardware;
- Testes a partir de ferramentas de simulação de circuitos, mais especificamente a ferramenta “Modelsim”;
- Testes em placa FPGA, mais especificamente Virtex 5 LX110T presente na plataforma *Xilinx* XUPV5-LX110T.

5.3. Cronograma

Tabela 3. Cronograma de Atividades para o Trabalho de Graduação 2

Tarefa	Nov/2011	Dez/2011	Jan/2012	Fev/2012	Mar/2012	Abril/2011	Mai/2012	Jun/2012	Jul/2012
Implementação dos Módulos	X	X	X	X					
Integração ao Decodificador			X	X	X				
Testes e Validação				X	X	X	X		
Documentação						X	X	X	X
Escrita do TG2							X	X	X
Defesa do TG2									X

6. Conclusão

A partir da realização do Trabalho de Graduação 1 foi possível obter base teórica sobre o assunto e se familiarizar com o ambiente de desenvolvimento do projeto. Também foi possível identificar o problema e estruturar táticas para resolvê-lo, baseando-se na arquitetura proposta e nos conceitos estudados.

Com base nesse trabalho foi obtido enfim os requisitos para possibilitar a realização do Trabalho de Graduação 2.

Referências

Berriel, E. A. (2005). O padrão de codificação de vídeo h.264 e uma implementação do módulo de predição intra quadro em hardware.

da Silva, L. I. L. (2003). Atos do poder executivo. decreto no 4.901.

Enéas, D. (2011). Otimização do módulo de predição intra-quadros e integração em um decodificador de vídeo para o sbtvd.

ITU (2007). Itu-t recommendation h.264. advanced video coding for generic audiovisual services. series h: Audiovisual and multimedia systems. infrastructure of audiovisual services - coding of moving video.

Lorencetti, M. A. (2010). Parser em vhdl para decodificador de vídeo h.264 para sbtvd.

Richardson, I. E. G. (2003). H.264 and mpeg-4 video compression:video coding for next-generation multimedia.