

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**DIERLES ENÉAS**

**OTIMIZAÇÃO DO MÓDULO DE PREDIÇÃO INTRA-  
QUADROS E INTEGRAÇÃO EM UM DECODIFICADOR DE  
VÍDEO PARA O SBTVD**

Porto Alegre

2011

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**OTIMIZAÇÃO DO MÓDULO DE PREDIÇÃO INTRA-  
QUADROS E INTEGRAÇÃO EM UM DECODIFICADOR DE  
VÍDEO PARA O SBTVD**

Projeto de Diplomação apresentado ao  
Departamento de Engenharia Elétrica da Universidade  
Federal do Rio Grande do Sul, como parte dos  
requisitos para Graduação em Engenharia Elétrica.

ORIENTADOR: Dr. Altamiro Amadeu Susin

COORIENTADOR: Dr. André Borin Soares

Porto Alegre

2011

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

DIERLES ENÉAS

**OTIMIZAÇÃO DO MÓDULO DE PREDIÇÃO INTRA-  
QUADROS E INTEGRAÇÃO EM UM DECODIFICADOR DE  
VÍDEO PARA O SBTVD**

Este projeto foi julgado adequado para fazer jus aos créditos da Disciplina de “Projeto de Diplomação”, do Departamento de Engenharia Elétrica e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: \_\_\_\_\_

Prof. Dr. Altamiro Amadeu Susin, UFRGS

Doutor pelo Institut National Polytechnique de Grenoble, França

Banca Examinadora:

Prof. Dr. Altamiro Amadeu Susin, UFRGS

Doutor pelo Institut National Polytechnique de Grenoble, França

Prof. Dr. Marcelo Götz, UFRGS

Doutor pela Universität Paderborn, Alemanha

Prof. Dr. Tiago Roberto Balen, UFRGS

Doutor pela Universidade Federal do Rio Grande do Sul, Brasil

Porto Alegre, julho de 2011.

## **DEDICATÓRIA**

Dedico este trabalho a toda minha família, em especial aos meus pais, que me deram todo o apoio, suporte e motivação necessários para ir em busca de meus objetivos, e ao meu irmão, que sempre me ajudou em diversos momentos difíceis do curso.

## **AGRADECIMENTOS**

Agradeço a toda equipe do LaPSI que me proporcionou um ótimo ambiente de trabalho para o desenvolvimento deste projeto.

Ao professor Altamiro Susin, pela orientação prestada ao longo deste trabalho.

Ao coorientador André Borin, por todos os esclarecimentos e conselhos essenciais para o avanço deste projeto.

A todos os participantes do projeto TV Digital, que de alguma forma contribuíram com seus ensinamentos.

À UFRGS e aos professores que foram fundamentais em minha formação.

À minha família, pelo apoio e motivação ao longo de todo o curso.

## RESUMO

A transmissão de vídeo digital com alta definição, nos sistemas de televisão digital, só é possível graças às técnicas de compressão de vídeo. Estas têm por objetivo minimizar a quantidade de dados necessária para a representação de uma sequência de vídeo.

Utilizado no Sistema Brasileiro de Televisão Digital (SBTVD), o padrão de compressão H.264 é, em geral, 50% mais eficiente, em termos de taxa de compressão, do que os padrões de compressão de vídeo anteriores [4]. Um dos motivos que levou a esta melhora foi a introdução da técnica de predição intra-quadros. Assim, mesmo os quadros que não possuem redundância temporal, isto é, semelhanças com quadros já decodificados, podem ser comprimidos através desta técnica.

Devido a limitações em termos de frequência, o módulo de predição intra-quadros, implementado no Laboratório de Processamento de Sinais e Imagens (LaPSI), não atendia a frequência mínima necessária para a decodificação de vídeo em *full HD*. Para atender este requisito do SBTVD, o módulo *intra* foi otimizado.

Este documento apresenta o módulo de predição intra-quadros, sua implementação, as otimizações, em termos de frequência, e a integração deste módulo em um decodificador de vídeo para o SBTVD, possibilitando a decodificação de vídeo com predição intra-quadros e inter-quadros.

**Palavras-chaves:** SBTVD, H.264, decodificador de vídeo, predição intra-quadros, predição inter-quadros, VHDL, FPGA.

## ABSTRACT

The transmission of digital video, in digital television systems, is only possible thanks to video compression techniques. These allow to minimize the amount of data required to represent a video sequence.

Present in the Brazilian Digital Television System (SBTV-D), H.264 compression standard is, usually, 50% more efficient, in terms of compression ratio, than previous video compression standards [4]. One of the reasons for this improvement was the introduction of intra-frame prediction technique. Thus, even the frames that have no temporal redundancy, i.e., have no similarities to previous decoded frames, can be compressed by this technique.

Due to limitations in terms of frequency, the intra-frame prediction module, implemented at the Laboratory of Signal and Image Processing (LaPSI), was not able to run in the minimum frequency required for full HD video decoding. To meet this SBTVD requirement, the *intra* module has been optimized.

This document presents the intra-frame prediction module, its implementation, the optimizations, in terms of frequency, and the integration of this module in a video decoder for SBTVD, which allow video decoding with intra-frame and inter-frame prediction.

**Keywords:** SBTVD, H.264, video decoder, intra-frame prediction, inter-frame prediction, VHDL, FPGA.

# SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>1</b>
<b>2. CONTEXTUALIZAÇÃO.....</b>	<b>2</b>
2.1 Sistema Brasileiro de Televisão Digital – SBTVD.....	2
2.2 Compressão de vídeo e o padrão H.264.....	3
2.2.1 Exploração das redundâncias.....	3
2.2.2 Espaços de cor: RGB e YCbCr.....	5
2.2.3 Sub-amostragem de cor.....	6
2.2.4 Macroblocos e blocos.....	7
2.3 O decodificador de vídeo para H.264.....	9
2.4 Transformadas.....	10
2.5 Quantização.....	11
2.6 Codificação de entropia.....	11
2.7 Filtro de deblocação.....	12
<b>3. PREDIÇÃO INTRA-QUADROS.....</b>	<b>13</b>
3.1 Algoritmo de predição intra-quadros.....	14
3.2 Derivação dos modos.....	16
3.3 Implementação do módulo de predição intra-quadros.....	18
3.3.1 Decodificação de contexto .....	19
3.3.2 Busca de vizinhanças.....	23
3.3.3 Geração das predições.....	23
3.3.4 Armazenamento das amostras.....	25
<b>4. OTIMIZAÇÕES NO MÓDULO DE PREDIÇÃO INTRA-QUADROS.....</b>	<b>28</b>
4.1 Aumento da frequência de funcionamento do módulo intra.....	28
4.1.1 Análise do caminho crítico .....	29
4.1.2 Inserção de registradores no caminho crítico.....	31
4.1.3 Re-engenharia do módulo de controle.....	32
4.2 Integração do módulo intra no decodificador de vídeo.....	34
4.2.1 Identificação do tipo de predição de cada macrobloco.....	35
4.2.3 Sincronização das amostras geradas.....	38
4.2.4 Situações de deadlock.....	39
<b>5. RESULTADOS.....</b>	<b>40</b>
<b>6. CONCLUSÕES.....</b>	<b>41</b>
<b>7. REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>42</b>



## LISTA DE FIGURAS

Figura 1: Redundância temporal.....	4
Figura 2: Redundância espacial.....	4
Figura 3: Sub-amostragem 4:4:4 [2].....	6
Figura 4: Sub-amostragem 4:2:2 [2].....	6
Figura 5: Sub-amostragem 4:2:0 [2].....	7
Figura 6: Ordem de codificação dos macroblocos em um quadro 176x144 (QCIF).....	8
Figura 7: Codificação em duplo Z para luminância e crominâncias .....	8
Figura 8: Decodificador de vídeo para o padrão H.264 [3].....	9
Figura 9: Estimação de movimento com precisão de 1 pixel e de 1/4 de pixel [2].....	9
Figura 10: Matriz de padrões e ordem de codificação para transformada DCT 4x4 [2].....	11
Figura 11: Predição de um bloco 4x4. As letras maiúsculas representam as vizinhanças [2].	13
Figura 12: Modos de predição 4x4 para luminância [2].....	13
Figura 13: Modos de predição 16x16 para luminância [2].....	14
Figura 14: Quadro original, à esquerda. Resultado da predição intra-quadros, à direita [2].	14
Figura 15: Algoritmo de predição intra-quadros para um macrobloco. Tracejado destaca decodificação de contexto.....	15
Figura 16: Ordem de geração das amostras em cada macrobloco [4].....	16
Figura 17: Bloco original, à esquerda. SAD para cada modo de predição, à direita [2].....	17
Figura 18: Arquitetura do módulo de predição intra-quadros [4].....	19
Figura 19: Arquitetura do decodificador de contexto [4].....	21
Figura 20: Arquitetura do módulo gerador de predições 4x4 (GP4x4) [6].....	24
Figura 21: Ordem de geração das predições de luminância e crominâncias. A cada ciclo, quatro amostras são geradas [4].....	24
Figura 22: Diagrama de tempo do módulo de predição intra-quadros.....	25
Figura 23: Amostras de luminância armazenadas em mem_viz (blocos azuis e verdes) .....	26
Figura 24: Denominação das vizinhanças 4x4, à esquerda. ....	27
Figura 25: Grupos de amostras que estão armazenados em v4x4 durante a predição de cada bloco [4].....	27
Figura 26: Blocos que influenciam no funcionamento do <i>intra</i> .....	29
Figura 27: Caminho crítico do módulo de predição intra-quadros.....	30
Figura 28: Localização do caminho crítico no decodificador de vídeo.....	31
Figura 29: Armazenamento das vizinhanças na versão do <i>intra</i> sem registradores.....	32
Figura 30: Armazenamento das vizinhanças na versão do <i>intra</i> com registradores.....	32
Figura 31: Alguns sinais de controle do módulo <i>intra</i> (situação 1).....	33
Figura 32: Alguns sinais de controle do módulo <i>intra</i> (situação 2).....	34
Figura 33: Arquitetura do módulo <i>parser</i> [3].....	36
Figura 34: Multiplexação dos módulos de predição através de MB_TP.....	37

## LISTA DE TABELAS

Tabela 1: Principais indicadores recebidos pelo bloco de predição intra-quadros.....	18
Tabela 2: Informações contidas no vetor de entrada <i>contexto</i> .....	19
Tabela 3: Informações contidas no vetor de saída <i>ctx</i> .....	20
Tabela 4: Organização da memória <i>mbm</i> .....	22
Tabela 5: Organização da memória <i>mem_viz</i> .....	26
Tabela 6: Requisitos mínimos para o funcionamento do decodificador de vídeo.....	28
Tabela 7: Desempenho das versões anterior e otimizada do <i>intra</i> .....	40
Tabela 8: Recursos do FPGA utilizados nas versões anterior e otimizada do <i>intra</i> .....	40

## LISTA DE ABREVIATURAS

ATSC	Advanced Television Systems Committee
AVC	Advanced Video Coding
BRAM	Block Random Access Memory
CABAC	Context-based Adaptive Binary Arithmetic Coding
CAVLC	Context-based Adaptive Variable Length Coding
DCT	Discrete Cosine Transform
DELET	Departamento de Engenharia Elétrica
DVB	Digital Video Broadcasting
FIFO	First In, First Out
FPGA	Field Programmable Gate Array
FINEP	Financiadora de Estudos e Projetos
HD	High Definition
HDTV	High Definition Television
ISDB-T	Integrated Services Digital Broadcasting Terrestrial
ITIQ	Inverse Transform, Inverse Quantization
LaPSI	Laboratório de Processamento de Sinais e Imagens
LUT	Look-Up Table
MC	Motion Compensation
MPEG	Moving Picture Experts Group
PAL	Phase Alternating Line
QCIF	Quarter Common Intermediate Format
RAM	Random Access Memory
RGB	Red, Green, Blue
SAD	Sum of Absolute Differences

SBTVD	Sistema Brasileiro de Televisão Digital
SD	Standard Definition
SDTV	Standard Definition Television
UFRGS	Universidade Federal do Rio Grande do Sul
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuits

# 1. INTRODUÇÃO

“Estimular a pesquisa e o desenvolvimento e propiciar a expansão de tecnologias brasileiras e da indústria nacional relacionadas à tecnologia de informação e comunicação” é um dos objetivos presentes no decreto N° 4.901 [1], assinado pelo então presidente Luiz Inácio Lula da Silva no dia 26 de Novembro de 2003. Tal decreto instituiu o Sistema Brasileiro de Televisão Digital (SBTVD) como padrão nacional a ser desenvolvido para, gradualmente, substituir o sistema de televisão analógico.

Como estímulo a pesquisa e ao desenvolvimento de tecnologias brasileiras, o Governo Federal financia, através de órgãos como o FINEP, diversos centros de pesquisas de universidades brasileiras. Entre estes está o Laboratório de Processamento de Sinais e Imagens (LaPSI) [7], centro de pesquisa do Departamento de Engenharia Elétrica da UFRGS (DELET), o qual está engajado no projeto de um decodificador de vídeo para o SBTVD.

Diversos blocos estão sendo desenvolvidos para a realização deste decodificador. Um desses é o módulo de predição intra-quadros, o qual possibilita a compressão espacial de dados, de acordo com o padrão H.264 [2]. Para que as características definidas pelo SBTVD sejam observadas, este módulo deve ser capaz de realizar a decodificação de vídeo em *full HD* (1920x1080 pixels por quadro). Atender este requisito, otimizando a frequência de funcionamento do bloco *intra*, e integrá-lo no decodificador de vídeo são os principais objetivos deste projeto.

Este documento inicia com a contextualização, explicando o ambiente e as bases necessárias para a compreensão do módulo de predição intra-quadros. O funcionamento e a implementação deste são detalhados no capítulo 3. O capítulo 4 apresenta as otimizações e a integração do módulo *intra* no decodificador de vídeo. Os resultados obtidos são expostos no capítulo 5. Finalmente, as conclusões e as perspectivas são apresentadas no capítulo 6.

## 2. CONTEXTUALIZAÇÃO

O Laboratório de Processamento de Sinais e Imagens, através do desenvolvimento do decodificador de vídeo, integra um projeto de âmbito nacional. Composta por centros de pesquisa de diversas universidades brasileiras, a rede H.264 Brasil, da qual o LaPSI faz parte, é responsável pelo desenvolvimento de um sistema completo de codificação e decodificação de vídeo e de áudio, atendendo aos requisitos estabelecidos pelo SBTVD.

### 2.1 Sistema Brasileiro de Televisão Digital – SBTVD

O Sistema Brasileiro de Televisão Digital (SBTVD), considerado o sistema de televisão digital mais avançado do mundo [8], tem como base o sistema japonês ISDB-T e, por esse motivo, também é conhecido como ISDB-Tb. Entre os motivos que justificam esta escolha estão: a qualidade e a robustez do sinal de vídeo/áudio, sendo esta última obtida através de técnicas de detecção e de correção de erros disponíveis neste sistema; o suporte à interatividade; e a recepção gratuita e de qualidade em aparelhos portáteis, como aparelhos celulares e tablets, inclusive quando estes estão em movimento. Além do padrão japonês, os padrões americano (ATSC) e europeu (DVB) também foram analisados.

Uma das características que diferencia o padrão brasileiro dos citados acima é a compressão de vídeo em MPEG-4 AVC, ou H.264. Isto permite uma maior capacidade de dados em um mesmo canal, quando comparada à compressão em MPEG-2, utilizada nos outros padrões.

Outra vantagem do padrão brasileiro é a possibilidade de interagir com a programação. Isto permite ao telespectador, por exemplo, ter acesso a bancos, participar de votações, fazer compras, etc. O software de interatividade - *middleware* Ginga - está sendo desenvolvido por instituições acadêmicas brasileiras e será mais robusto do que o utilizado no padrão japonês.

Além disso, o SBTVD suporta a transmissão de mais de um programa em um mesmo canal. Tal característica, conhecida como multiprogramação, possibilita que um único canal apresente:

- Um programa em altíssima definição, ou *Full HD* (1920x1080);
- Dois programas em alta definição, ou *HD* (1280x720);
- Quatro programas em definição padrão, ou *SD* (720x480);
- Um programa em *HD* e dois programas em *SD*.

## **2.2 Compressão de vídeo e o padrão H.264**

No sistema brasileiro de televisão, cada canal disponibiliza uma banda de 6 MHz para a transmissão das informações de vídeo/áudio. Processos de modulação adequados permitem uma taxa em torno de 20 Mbits/segundo para a transmissão de um sinal digital nesta banda. Para uma transmissão HDTV, por exemplo, a taxa necessária para um vídeo sem compressão seria superior a 1 Gbit/segundo, mais de 50 vezes maior do que a taxa permitida.

A necessidade de compressão também fica evidente para o armazenamento de dados. Duas horas de vídeo com definição SDTV sem compressão exigiriam aproximadamente 240 Gbytes de memória. Menos de 1 Gbyte é suficiente para o mesmo vídeo comprimido.

Reduzir a quantidade de informações para codificar a mesma sequência de vídeo é possível graças à exploração de redundâncias temporais, espaciais e de frequência.

### **2.2.1 Exploração das redundâncias**

A figura 1 ilustra um exemplo de redundância temporal através de dois quadros consecutivos. Percebe-se que a maior parte da imagem não sofreu alterações. Além disso, nota-se que houve um movimento vertical em um dos objetos. Tal movimento pode ser

caracterizado com o auxílio de *vetores de movimento*, os quais apontam em direção à posição anterior dos pixels (ou frações de pixel, como será visto posteriormente). A técnica de predição inter-quadros é utilizada para compressão em sequências de vídeo com redundâncias temporais.



Figura 1: Redundância temporal

Na figura 2 observa-se algumas regiões homogêneas destacadas, caracterizando a presença de redundância espacial. Desta forma, a compressão pode ser obtida através da técnica de predição intra-quadros, detalhada no capítulo 3.

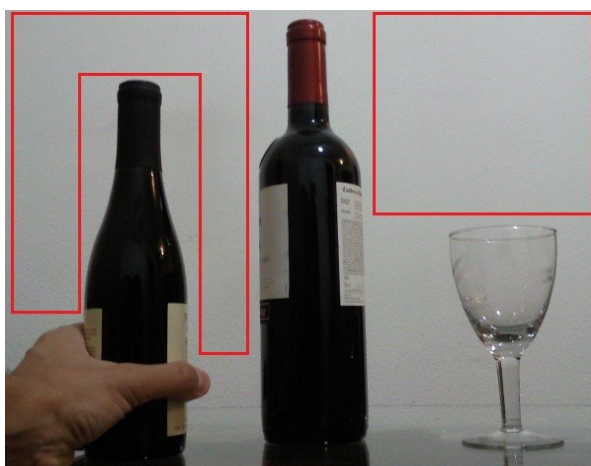


Figura 2: Redundância espacial



Já a redundância de frequência é decorrente da maior sensibilidade da visão humana às baixas frequências. Desta forma, pode-se desprezar as informações de amplitudes baixas em altas frequências – utilizando um parâmetro de quantização (2.5) – sem haver perdas significativas na qualidade da imagem.

### **2.2.2 Espaços de cor: *RGB* e *YCbCr***

Na codificação de imagens monocromáticas basta a informação de luminância (*Y*) para a representação de cada pixel. Já para imagens coloridas, há a necessidade de três componentes por pixel.

No espaço *RGB*, cada componente representa uma das cores primárias aditivas: vermelho (*R*), verde (*G*) e azul (*B*). A soma destas três cores em diferentes proporções possibilita a formação de todas as cores visíveis para os humanos.

Já no espaço *YCbCr*, cada pixel é formado por uma componente de luminância e duas componentes de croma. Como a luminância é obtida através de uma média ponderada das três componentes de cor, basta a informação de duas componentes de croma para se deduzir a terceira. Desta forma, apenas as componentes de croma de azul (*Cb*) e de vermelho (*Cr*) são informadas.

Como a visão humana apresenta maior sensibilidade à luminância, o espaço *YCbCr* apresenta uma grande vantagem em termos de compressão. Isto ocorre devido à possibilidade de sub-amostrar as cores.

### **2.2.3 Sub-amostragem de cor**

Três padrões normalmente são utilizados para a sub-amostragem de cor: 4:4:4, 4:2:2 e 4:2:0.

Na sub-amostragem 4:4:4 (figura 3), a cada amostra de luminância há uma amostra de Cb e outra de Cr. Desta forma, não há nenhuma vantagem em relação ao espaço RGB.

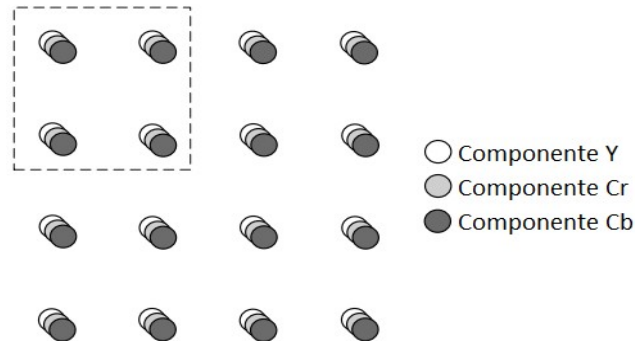


Figura 3: Sub-amostragem 4:4:4 [2]

Na sub-amostragem 4:2:2 (figura 4), a cada duas amostras de luminância, no sentido horizontal, há uma amostra de Cb e outra de Cr. Assim, caso a frequência de amostragem de luminância for 13,5 MHz, a frequência de amostragem das crominâncias será 6,75 MHz.

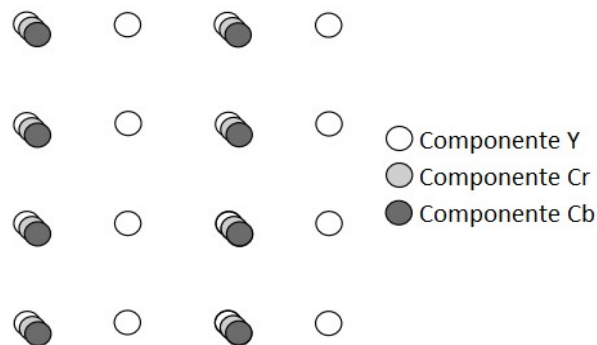


Figura 4: Sub-amostragem 4:2:2 [2]

Utilizada na maior parte das aplicações do padrão H.264, a sub-amostragem 4:2:0 (figura 5) apresenta uma amostra de Cb e outra de Cr para cada duas amostras de luminância,

nos sentidos horizontal e vertical. Assim, pode-se dizer que cada pixel é representado por 1,5 amostra. Isto representa um ganho de 50% em relação ao espaço RGB (3 amostras por pixel).

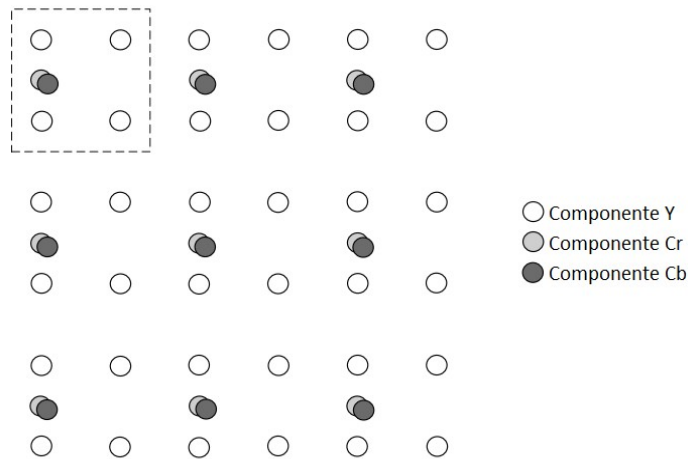


Figura 5: Sub-amostragem 4:2:0 [2]

#### 2.2.4 Macroblocos e blocos

No padrão H.264, cada quadro é dividido em **macroblocos** de 16x16 pixels. No caso da sub-amostragem 4:2:0, cada macrobloco é composto por um conjunto de 16x16 amostras de luminância e um conjunto de 8x8 amostras para cada componente de crominância (Cb e Cr).

Cada macrobloco, por sua vez, é subdividido em **blocos** de 4x4 amostras de luminância e 2x2 amostras de crominâncias.

A ordem de codificação dos macroblocos dentro de um quadro é da esquerda para a direita, de cima para baixo. A figura 6 mostra esta codificação para o formato QCIF (176x144), muito utilizado em aparelhos celulares. Já a figura 7 ilustra a ordem de codificação de cada bloco, conhecida como codificação em duplo Z.

0	1	2	3	4	5	6	7	8	9	10
11	12	13	14	...	...					
⋮										
⋮										
										⋮
										⋮
							...	...	97	98

Figura 6: Ordem de codificação dos macroblocos em um quadro 176x144 (QCIF)

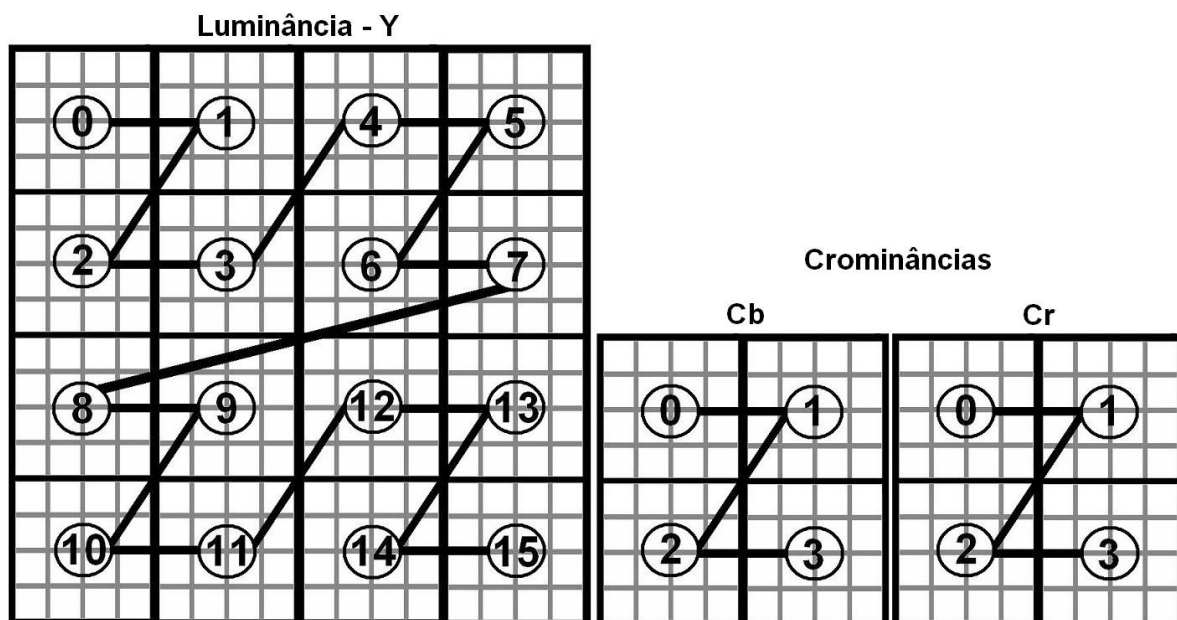


Figura 7: Codificação em duplo Z para luminância e crominâncias dos blocos 4x4 de um macrobloco

### 2.3 O decodificador de vídeo para H.264

Segundo o padrão H.264, a maior parte das informações contidas em uma imagem são preditas através das informações já decodificadas, seja de diferentes quadros (predição inter-quadros), ou do quadro atual (predição intra-quadros). A figura 8 mostra o esquemático de um decodificador de vídeo para o padrão H.264.

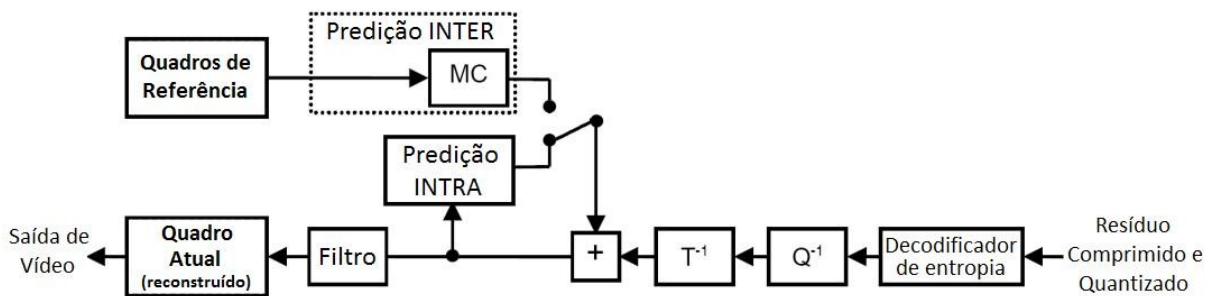


Figura 8: Decodificador de vídeo para o padrão H.264 [3]

No caso da **predição inter-quadros**, a imagem atual é reconstruída através de um módulo de compensação de movimento, MC (do inglês: *Motion Compensation*), o qual toma quadros passados ou futuros como referência para estimar o quadro atual. Esta estimativa é feita através de vetores de movimento. No padrão H.264, estes vetores têm precisão de até  $\frac{1}{4}$  de pixel, como mostra a figura 9. Neste caso, os vetores são obtidos por interpolação das amostras.

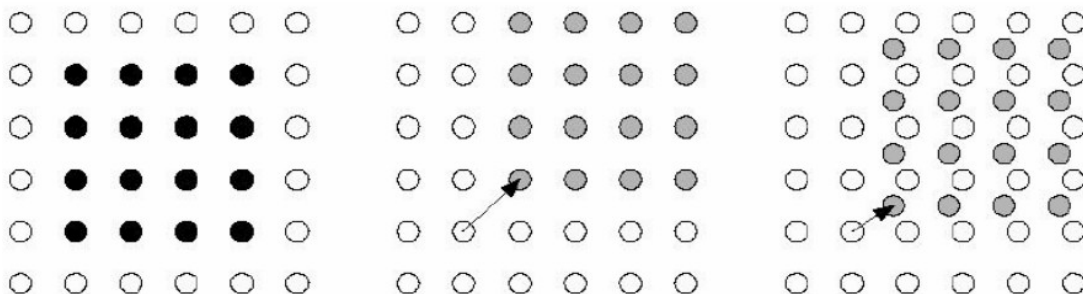


Figura 9: Estimativa de movimento com precisão de 1 pixel e de 1/4 de pixel [2]

Já na **predição intra-quadros**, a imagem é predita gradualmente, utilizando como referência as vizinhanças previamente decodificadas.

Os módulos de predição inter e intra-quadros estão presentes tanto no codificador como no decodificador de vídeo. Sua presença no codificador é necessária para calcular a diferença entre a imagem original e a imagem predita. Esta diferença, conhecida como resíduo, é transformada (2.4) e quantizada (2.5) para que se tenha a máxima compressão de dados.

No decodificador, o resíduo passa pelos módulos de decodificação de entropia (2.6), quantização inversa e transformada inversa, para finalmente ser somado à saída de um dos módulos de predição. A saída de vídeo será obtida passando o resultado desta soma através de um filtro de deblocação (2.7).

## **2.4 Transformadas**

A técnica de transformada do domínio espacial para o domínio de frequências espaciais é mais um processo para a compressão de dados. Aplicadas sobre os resíduos, as transformadas DCT e Haddamard são utilizadas no padrão H.264, sendo esta última aplicada aos coeficientes DC. A figura 10 ilustra a matriz de padrões para a transformada DCT 4x4, assim como a ordem de codificação dos coeficientes após o processo de quantização. Percebe-se que os primeiros coeficientes representam baixas frequências, ao passo que os últimos contêm as informações de altas frequências. Isto permitirá ao codificador de entropia reduzir o número de bits necessários para a codificação destes, visto que, após passar pela etapa de quantização, grande parte dos coeficientes de altas frequências são nulos se apresentarem amplitude inferior ao passo de quantização. Desta forma, aplica-se a exploração de redundância de frequência.

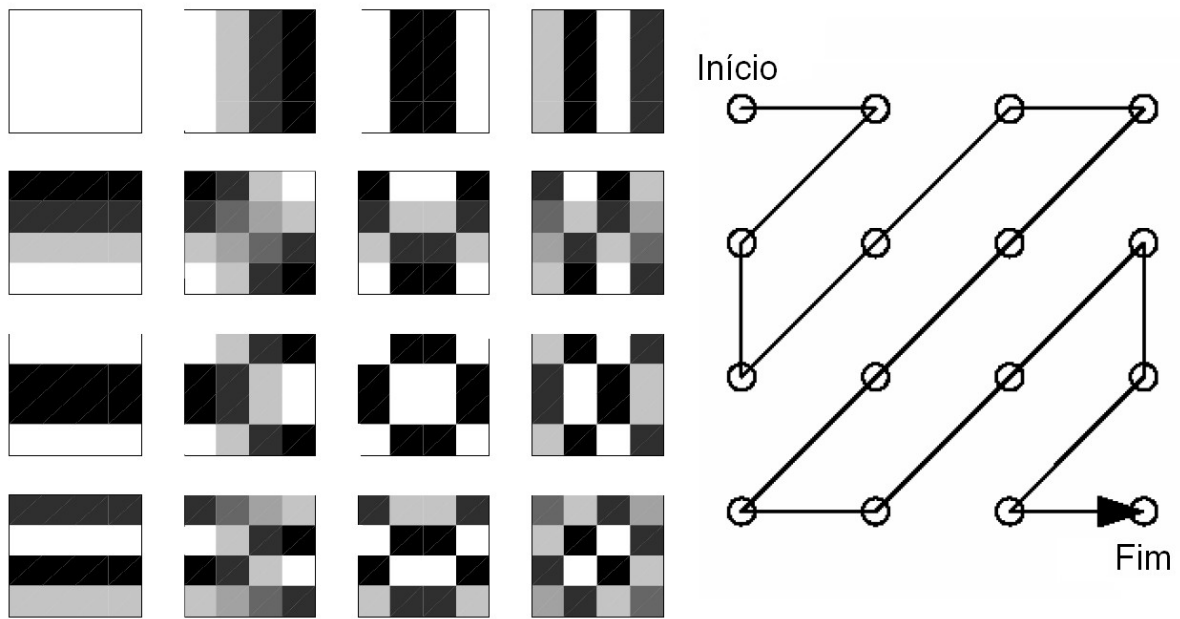


Figura 10: Matriz de padrões e ordem de codificação para transformada DCT 4x4 [2]

## 2.5 Quantização

A quantização é realizada logo após a aplicação das transformadas. Todos os coeficientes são quantizados através de um parâmetro responsável por definir a taxa de bits na saída do codificador. Este parâmetro estará presente na sequência de bits recebida pelo decodificador, conhecida como *bitstream*. No padrão H.264, a quantização é o único processo que introduz perdas na qualidade da imagem.

Depois de quantizados, os coeficientes são enviados ao codificador de entropia.

## 2.6 Codificação de entropia

O codificador de entropia se aproveita do fato que grande parte dos coeficientes quantizados são nulos para gerar o *bitstream* com a menor quantidade de bits possível. Duas técnicas são utilizadas em H.264: Codificação de comprimento variável (CAVLC) e Codificação Aritmética (CABAC). Ambas são adaptativas ao contexto, isto é, dependem do

elemento a ser codificado, da fase em que o algoritmo de codificação se encontra e dos elementos sintáticos que já foram codificados.

## **2.7 Filtro de deblocagem**

O filtro de deblocagem serve para minimizar os efeitos causados pela decodificação de blocos. Em H.264, tal filtro deve ser capaz de diferenciar uma aresta real, presente na imagem original, de uma “aresta” resultante de um alto parâmetro de quantização. Desta forma, melhora-se a qualidade subjetiva da imagem.



### 3. PREDIÇÃO INTRA-QUADROS

A predição intra-quadros é baseada nas vizinhanças. Cada bloco é predito através de uma média ponderada dos valores dos pixels à esquerda e acima do bloco, de acordo com o **modo** de predição. Este define a direção em que os pixels vizinhos são propagados causando o menor erro e, conseqüentemente, o menor resíduo. A figura 11 define os vizinhos que são utilizados para a predição de um bloco 4x4. Já a figura 12 mostra os nove modos possíveis para a predição Intra 4x4 para luminância.

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 11: Predição de um bloco 4x4. As letras maiúsculas representam as vizinhanças [2]

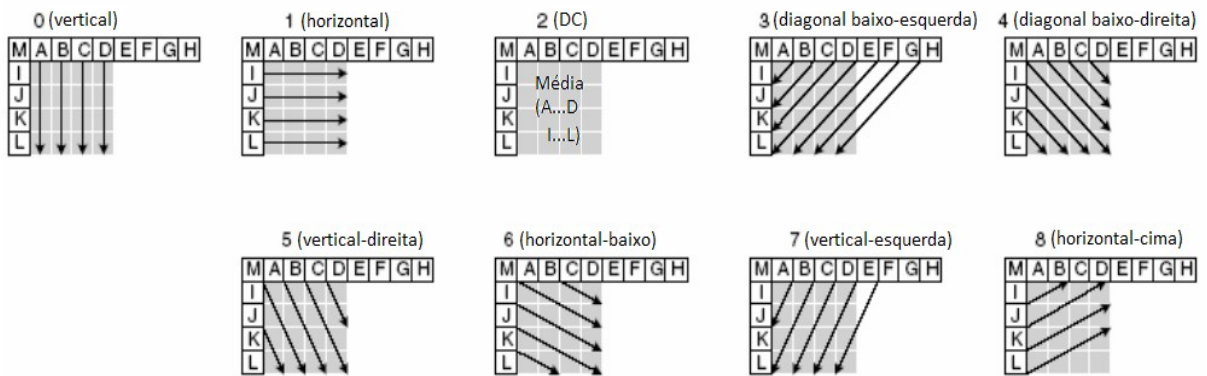


Figura 12: Modos de predição 4x4 para luminância [2]

No modo 3 de predição 4x4 para luminância, por exemplo, os pixels *d*, *g*, *j* e *m* são obtidos pela expressão  $(D+2E+F)/4$ .

Na predição Intra 16x16 existem apenas quatro modos (figura 13). Os mesmos são utilizados para a predição das componentes de crominâncias. Neste caso, porém, o modo DC é o modo 0 e o modo 2 é o modo vertical.

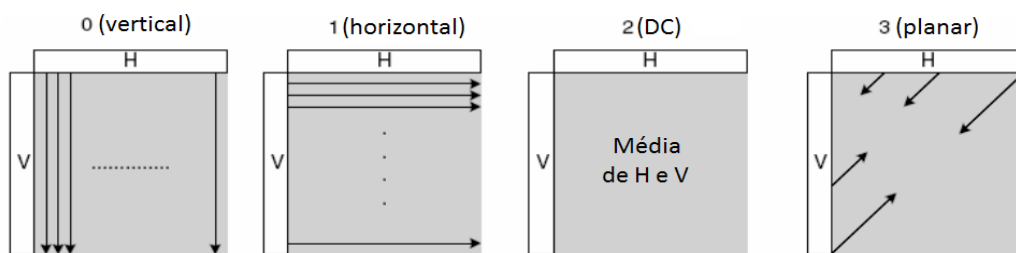


Figura 13: Modos de previsão 16x16 para luminância [2]

A figura 14 mostra o resultado da previsão intra-quadros de um quadro, desconsiderando a soma dos resíduos após a geração de cada bloco.



Figura 14: Quadro original, à esquerda. Resultado da previsão intra-quadros, à direita [2]

### 3.1 Algoritmo de previsão intra-quadros

O primeiro passo da previsão intra-quadros é verificar o tipo de previsão: 4x4 ou 16x16. A seguir, verifica-se a disponibilidade dos pixels vizinhos. No caso da primeira coluna de macroblocos não há disponibilidade dos macroblocos vizinhos à esquerda. Já a primeira linha de macroblocos não possui vizinhança superior. Caso haja disponibilidade de vizinhos e estes não estejam com a permissão para previsão intra-quadros negada (ver indicador *constrained\_intra\_pred\_flag* na tabela 1), derivam-se os modos de previsão seguindo o algoritmo definido no subcapítulo 3.2. Estas etapas, que precedem a geração das previsões, fazem parte da decodificação do *contexto* (ver 3.3.1).

É importante destacar que, após a predição de cada bloco, soma-se o resultado desta ao resíduo, antes de realizar a predição do bloco seguinte. Desta forma, não há propagação de erros.

A figura 15 mostra o algoritmo de predição intra-quadros para um macrobloco. Na implementação, optou-se por realizar a busca e geração das amostras na seguinte ordem: luminância, crominância Cb e, por último, crominância Cr. Esta ordem está representada na figura 16. Neste caso, a busca das amostras vizinhas de luminância é feita paralelamente à decodificação do contexto.

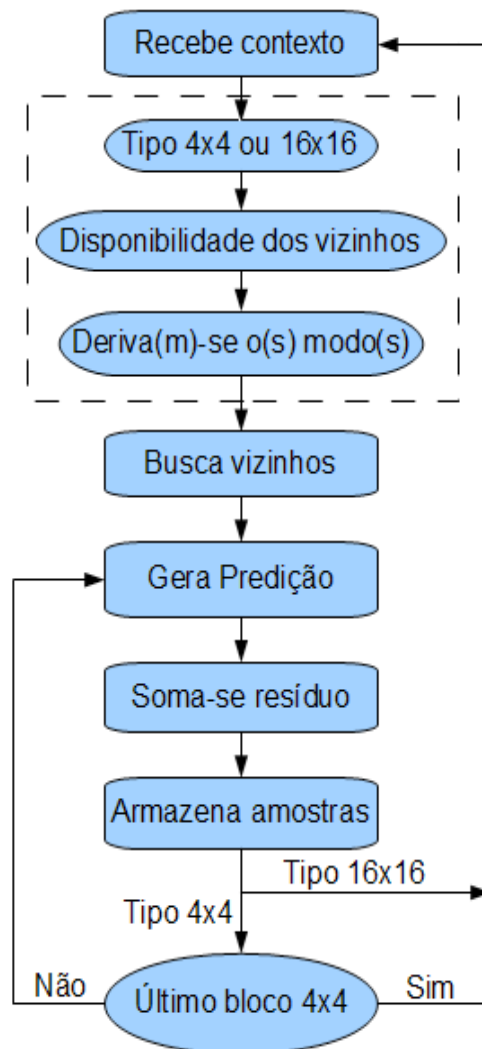


Figura 15: Algoritmo de predição intra-quadros para um macrobloco. Tracejado destaca decodificação de contexto

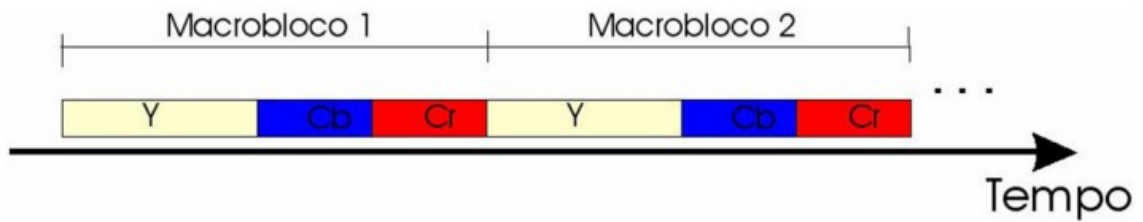


Figura 16: Ordem de geração das amostras em cada macrobloco [4]

### 3.2 Derivação dos modos

Sendo  $idx$  o índice do bloco do qual o modo será calculado,  $A$  o seu vizinho à esquerda,  $B$  o bloco superior,  $prev\_intra4x4\_pred\_mode\_flag$  o indicador de que utilizar-se-á o modo predito e  $rem\_intra4x4\_pred\_mode$  um vetor com os modos que podem ser utilizados quando não há predição de modo, tem-se o seguinte algoritmo para derivação dos modos:

**Para  $idx$  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  $pred\_intra4x4\_pred\_mode\_flag(idx) = '1'$**

**modo(idx) = pmodo;**

**Senão**

**Se  $rem\_intra4x4\_pred\_mode(idx) < pmodo$**

**modo(idx) =  $rem\_intra4x4\_pred\_mode\_flag(idx)$ ;**

**Senão**

**modo(idx) =  $rem\_intra4x4\_pred\_mode\_flag(idx) + 1$ ;**

**Repete para cada um dos blocos.**

Tanto  $pred\_intra4x4\_pred\_mode\_flag$  como  $rem\_intra4x4\_pred\_mode$ , definidos pela norma H.264, são gerados pelo codificador de vídeo. Grande esforço computacional é

requerido para o cálculo destes, visto que o modo de predição deve ser aquele que minimiza o valor do resíduo. A figura 17 exemplifica, para um bloco 4x4, o cálculo do modo de predição para se obter a menor soma absoluta entre as diferenças do bloco original e do bloco predito. Este método é conhecido como SAD (do inglês: “Sum of Absolute Differences”).

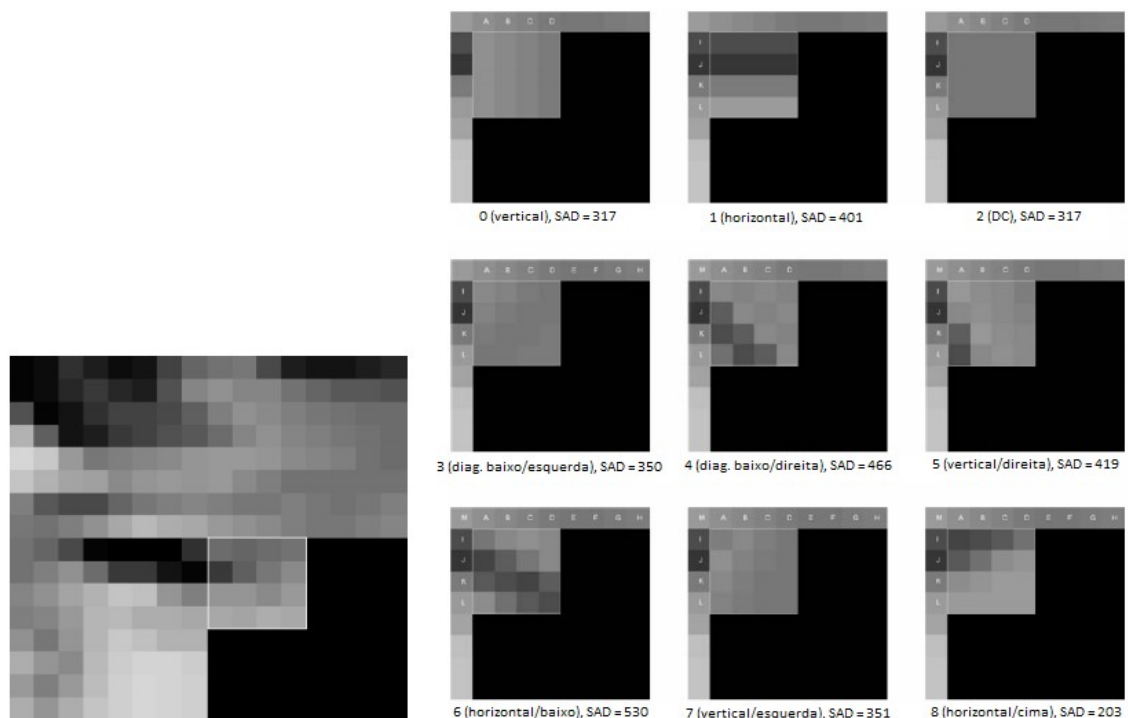


Figura 17: Bloco original, à esquerda. SAD para cada modo de predição, à direita [2]

Percebe-se que o modo 8 é o que minimiza o erro pelo método SAD. Desta forma, o codificador deverá gerar *pred\_intra4x4\_pred\_mode\_flag* e *rem\_intra4x4\_pred\_mode* de maneira que este seja o modo de predição para este bloco.

É importante destacar que este algoritmo é válido apenas para predição intra 4x4 de luminâncias. No caso da predição 16x16, o modo é definido, na norma, pelo indicador *mb\_type*, o qual também indica o tipo de predição (16x16 ou 4x4). Por esse motivo, este indicador foi denominado *tipo\_modo* na implementação do módulo *intra*. Para a predição de crominâncias, o modo é extraído do indicador *intra\_chroma\_pred\_mode*.

### 3.3 Implementação do módulo de predição intra-quadros

O módulo de predição intra-quadros foi desenvolvido através da linguagem de descrição de hardware VHDL para ser implementado no FPGA Virtex-II Pro VP70, da *Xilinx*. A implementação foi realizada pelo engenheiro Eduardo Agostini Berriel no Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande do Sul [4]. Posteriormente, a implementação passou a ser realizada no FPGA Virtex 5 LX110T, o qual está presente na plataforma XUPV5-LX110T, também da *Xilinx*.

A tabela 1 descreve os principais indicadores, descritos pela norma H.264 [5], que são recebidos pelo bloco de predição intra-quadros. O nome utilizado para cada indicador na implementação é indicado entre parênteses.

**Tabela 1: Principais indicadores recebidos pelo bloco de predição intra-quadros**

Indicador	Nº de bits	Descrição
mb_type ( <i>tipo_modo</i> )	5	Informa o tipo de predição luma e o modo para tipo 16x16.
prev_intra4x4_pred_mode_flag ( <i>fmodo</i> )	16	Vetor com 16 flags que indicam quais blocos 4x4 tem o seu modo derivado por predição.
rem_intra4x4_pred_mode ( <i>vmodo</i> )	48	Vetor com 16 modos entre 0 e 7 dos blocos 4x4 que não têm o seu modo derivado por predição.
intra_chroma_pred_mode ( <i>cmodo</i> )	2	Modo de predição das crominâncias. 0 para DC, 1 para Horizontal, 2 para Vertical e 3 para Planar.
mb_addr ( <i>addr</i> )	16	Endereço do macrobloco no quadro.
pic_width_in_mbs ( <i>larg + 1</i> )	8	Largura da tela em macroblocos.
constrained_intra_pred_flag ( <i>nao_usar</i> )	1	Informa se um macrobloco não pode ser utilizado para predição intra-quadros de seus vizinhos.
chroma_format_idc ( <i>cor</i> )	2	Informa a sub-amostragem de cor. 0 para monocromático, 1 para 4:2:0, 2 para 4:2:2 e 3 para 4:4:4.

A arquitetura desenvolvida para o bloco de predição intra-quadros é mostrada na figura 18. Tal arquitetura pode ser subdividida em quatro partes, responsáveis pelas seguintes tarefas: decodificação de contexto, busca de vizinhanças, geração das predições e

armazenamento das amostras que correspondem à soma da predição com o resíduo. Cada uma destas partes, assim como seus respectivos blocos, são detalhados a seguir.

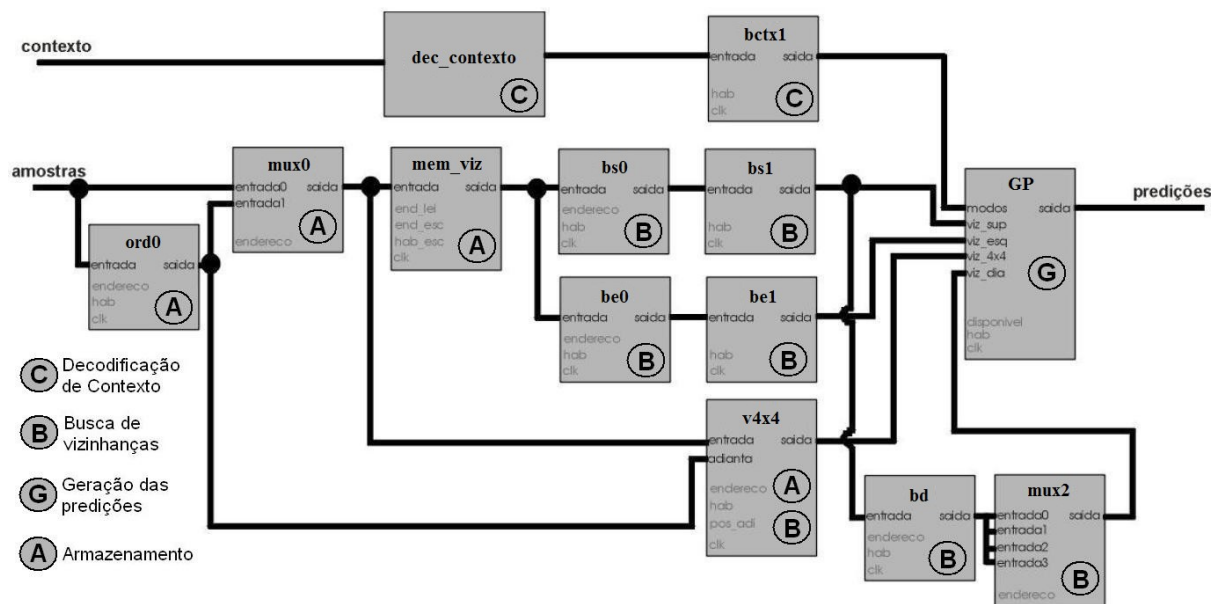


Figura 18: Arquitetura do módulo de predição intra-quadros [4]

### 3.3.1 Decodificação de contexto

O decodificador de contexto é o módulo responsável pela interpretação dos dados provenientes de outros blocos presentes no decodificador de vídeo. Estes dados formam o vetor de entrada *contexto*, o qual contém as informações listadas na tabela 2. A partir destas informações, a decodificação de contexto deve informar a disponibilidade das vizinhanças, o tipo de predição (4x4 ou 16x16 para predição intra-quadros), assim como os modos de predição para luminância e crominâncias. A saída do decodificador de contexto, a qual contém estas informações, foi denominada *ctx* (ver tabela 3).

Tabela 2: Informações contidas no vetor de entrada *contexto*

Vetor de Entrada: <i>contexto</i>	Nº de bits	Posição	Descrição
col	8	107 - 100	Posição do macrobloco na linha.
addr	16	99 - 84	Posição do macrobloco no quadro.

larg	8	83 - 76	Largura da tela em macroblocos menos 1.
cor	2	75 - 74	Informa a sub-amostragem de cor.
tipo_mod	5	73 - 69	Informa se o tipo de predição luma e o modo para tipo 16x16.
fmodo	16	68 - 53	Vetor com 16 flags que indicam quais blocos 4x4 tem o seu modo derivado por predição.
vmodo	48	52 - 5	Vetor com 16 modos entre 0 e 7 dos blocos 4x4 que não têm o seu modo derivado por predição.
cmodo	2	4 - 3	Modo de predição das crominâncias.
nao_usar	1	2 - 2	Informa se o macrobloco não pode ser utilizado para predição intra-quadros de seus vizinhos.
ff	2	1 - 0	Informa se o quadro é quadro/campo adaptativo. Na norma é definido como <i>mbafframeflag</i> .

**Tabela 3: Informações contidas no vetor de saída *ctx***

<b>Vetor de saída: <i>ctx</i></b>	<b>Nº de bits</b>	<b>Posição</b>	<b>Descrição</b>
disp_esq	1	75 - 75	Vizinhos pertencentes ao macrobloco à esquerda estão disponíveis.
disp_sup	1	74 - 74	Vizinhos pertencentes ao macrobloco superior estão disponíveis.
disp_sd	1	73 - 73	Vizinhos pertencentes ao macrobloco à direita do macrobloco superior estão disponíveis.
disp_se	1	72 - 72	Vizinhos pertencentes ao macrobloco à esquerda do macrobloco superior estão disponíveis.
nao_usar	1	71 - 71	Informa se um macrobloco não pode ser utilizado para predição intra-quadros de seus vizinhos.
reservado	1	70 - 70	Reservado para possíveis modificações.
tipo	2	69 - 68	Não é intra (00), Intra 16x16 (10), Intra 4x4 (11).
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.

A figura 19 mostra a arquitetura do decodificador de contexto.

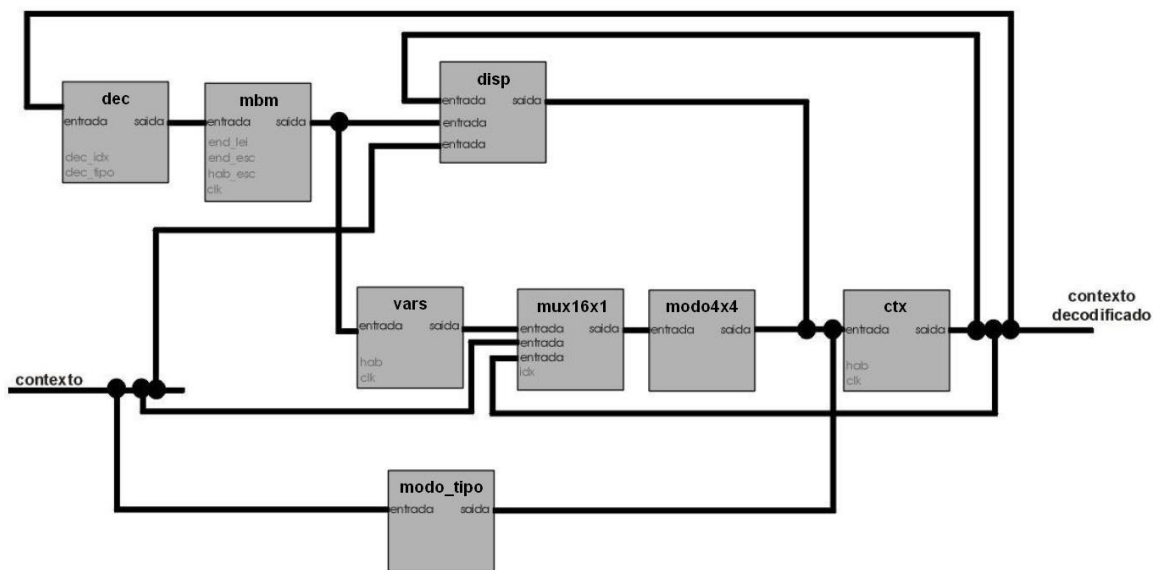


Figura 19: Arquitetura do decodificador de contexto [4]

A função de cada bloco é definida a seguir:

- **dec:** multiplexador responsável por selecionar os modos 4x4, o indicador *nao\_usar* e o tipo de cada macrobloco, disponibilizando-os um a um à memória **mbm**.
- **mbm:** memória (2.048 palavras de 4 bits) que grava os dados provenientes de **dec** para a última linha de macroblocos. No caso de um quadro em *Full HD* (1920x1080), por exemplo, esta linha é composta pelos últimos 120 macroblocos. A tabela 4 mostra como esta memória é implementada.

**Tabela 4: Organização da memória mbm**

<b>Nº de palavras de 4 bits</b>	<b>Posição</b>	<b>Descrição</b>
1920	0 - 1919	Modos 4x4 da linha de macroblocos. Cada macrobloco precisa de 16 palavras de 4 bits para representar os modos de seus blocos 4x4.
120	1922 - 2041	Indicadores “ <i>nao_usar</i> ”(3) e “ <i>tipo</i> ”(1 downto 0) de cada macrobloco da linha de macroblocos.

- **disp**: calcula a disponibilidade dos macroblocos vizinhos.
- **vars**: memória que contém o tipo e o modo dos macroblocos superior e à esquerda do macrobloco atual.
- **mux16x1**: multiplexador responsável pela seleção de todas as informações necessárias para a realização da derivação dos modos de predição de cada bloco 4x4: disponibilidade, tipo e modo dos blocos vizinhos, além dos indicadores *fmodo* e *vmodo*.
- **modo4x4**: bloco responsável pela derivação dos modos de predição de cada bloco 4x4 através do algoritmo apresentado no subcapítulo 3.2.
- **modo\_tipo**: decodifica o tipo (16x16 ou 4x4) e o modo de predição 16x16.
- **ctx**: buffer com o *contexto* decodificado para o macrobloco atual.

Após o término da etapa de decodificação de contexto, o vetor de saída *ctx* é chaveado através do módulo *bctx1*, dando início à geração das predições para o presente macrobloco.

### **3.3.2 Busca de vizinhanças**

Os módulos **bs0** e **be0** são responsáveis pela busca das 16 amostras acima e à esquerda do macrobloco atual, as quais estão armazenadas na memória **mem\_viz**. Já **bd** contém a

amostra da diagonal, isto é, a amostra  $M$  (ver figura 11) que pode ser utilizada para a predição do bloco 0. Quando as amostras externas ao macrobloco atual estão armazenadas em  $bs0$  e  $be0$ , os módulos  $bs1$  e  $be1$  as chaveiam para o módulo gerador de predições (**GP**). No caso da predição do bloco 5 (ver figura 7), as amostras  $E$ ,  $F$ ,  $G$  e  $H$  são buscadas diretamente na memória  $mem\_viz$  pelo módulo GP. O multiplexador **mux2** seleciona a amostra de luminância, cromaância Cb ou cromaância Cr para as amostras da diagonal.

O módulo **v4x4** armazena e fornece as amostras internas do macrobloco atual, as quais fazem parte da vizinhança dos blocos 4x4. Tanto **v4x4** como  $mem\_viz$  são detalhados no subcapítulo 3.3.4.

### 3.3.3 Geração das predições

O módulo GP é responsável pela geração das predições. Este é formado por dois grandes blocos puramente combinacionais: GP\_Luma16 e GP4x4. O primeiro realiza as predições do tipo 16x16 e o segundo, cuja arquitetura é mostrada na figura 20, é responsável pelas predições 4x4.

A cada ciclo, quatro predições são geradas. Cada bloco **PE**, presente na arquitetura do módulo GP4x4, é responsável pelo cálculo da predição de uma amostra. Na entrada destes blocos,  $n1$ ,  $n2$  e  $n3$  representam as amostras vizinhas ao bloco atual;  $c1$ ,  $c2$  e  $c3$  representam os coeficientes que multiplicam estas amostras de acordo com o modo de predição. No caso do modo DC, uma lógica auxiliar é utilizada.

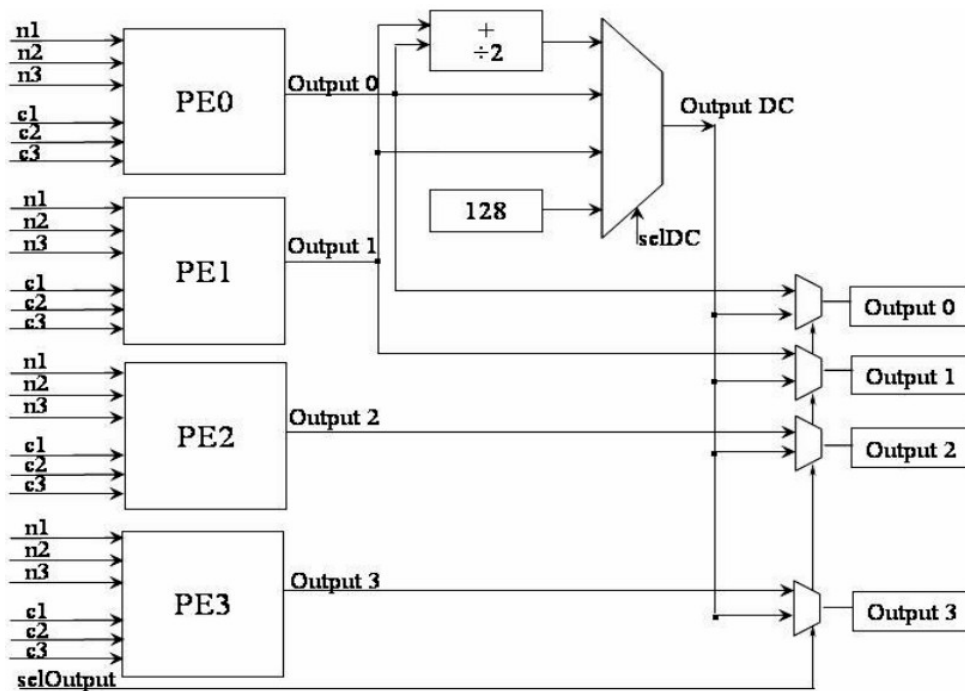


Figura 20: Arquitetura do módulo gerador de previsões 4x4 (GP4x4) [6]

A ordem em que as previsões são geradas é mostrada na figura 21. As primeiras previsões de luminância do macrobloco atual são geradas quatro ciclos após o chaveamento do contexto decodificado.

Luminância															
0	1	2	3	16	17	18	19	64	65	66	67	80	81	82	83
4	5	6	7	20	21	22	23	68	69	70	71	84	85	86	87
8	9	10	11	24	25	26	27	72	73	74	75	88	89	90	91
12	13	14	15	28	29	30	31	76	77	78	79	92	93	94	95
32	33	34	35	48	49	50	51	96	97	98	99	112	113	114	115
36	37	38	39	52	53	54	55	100	101	102	103	116	117	118	119
40	41	42	43	56	57	58	59	104	105	106	107	120	121	122	123
44	45	46	47	60	61	62	63	108	109	110	111	124	125	126	127
128	129	130	131	144	145	146	147	192	193	194	195	208	209	210	211
132	133	134	135	148	149	150	151	196	197	198	199	212	213	214	215
136	137	138	139	152	153	154	155	200	201	202	203	216	217	218	219
140	141	142	143	156	157	158	159	204	205	206	207	220	221	222	223
160	161	162	163	176	177	178	179	224	225	226	227	240	241	242	243
164	165	166	167	180	181	182	183	228	229	230	231	244	245	246	247
168	169	170	171	184	185	186	187	232	233	234	235	248	249	250	251
172	173	174	175	188	189	190	191	236	237	238	239	252	253	254	255

Crominâncias															
0	1	2	3	16	17	18	19								
4	5	6	7	20	21	22	23								
8	9	10	11	24	25	26	27								
12	13	14	15	28	29	30	31								
32	33	34	35	48	49	50	51								
36	37	38	39	52	53	54	55								
40	41	42	43	56	57	58	59								
44	45	46	47	60	61	62	63								

Figura 21: Ordem de geração das previsões de luminância e crominâncias.

A cada ciclo, quatro amostras são geradas [4]

A decodificação do contexto, a partir do segundo macrobloco, acontece paralelamente à geração das predições de crominância do macrobloco anterior. Isto pode ser observado no diagrama de tempos do módulo de predição intra-quadros (figura 22).

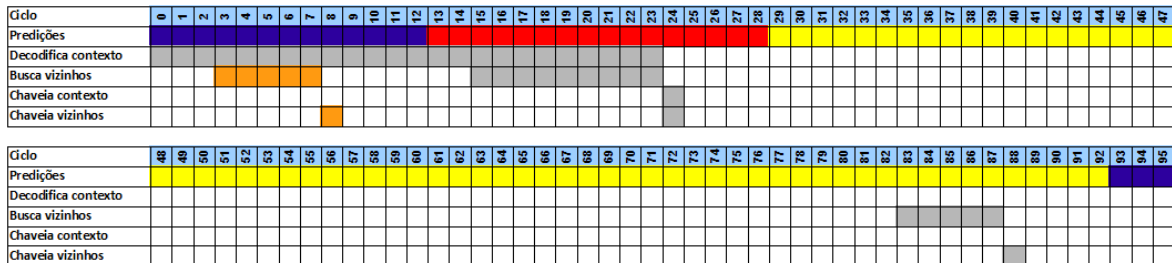


Figura 22: Diagrama de tempo do módulo de predição intra-quadros

Neste diagrama, blocos amarelos representam amostras de luminância, blocos azuis correspondem às amostras de crominância Cb e blocos vermelhos representam as amostras de crominância Cr. Os blocos em cinza indicam o que está acontecendo desde a decodificação de contexto do primeiro macrobloco. Já os blocos em laranja ocorrem a partir da decodificação do contexto do segundo macrobloco. Estes representam operações relativas ao macrobloco anterior. Da mesma forma, as amostras de crominâncias geradas nos ciclos 0 a 28 correspondem ao macrobloco anterior, cujo contexto já foi decodificado e chaveado.

### 3.3.4 Armazenamento das amostras

Assim que as predições são geradas e somadas ao resíduo, as amostras resultantes devem ser memorizadas. Aquelas que fazem parte das vizinhanças entre macroblocos são armazenadas em `mem_viz`. Já as que fazem parte das vizinhanças dos blocos 4x4 são armazenadas em `v4x4`. Na figura 23, os blocos em azul e em verde representam as amostras de luminância armazenadas em `mem_viz`. Já os blocos em amarelo representam as amostras de luminância provenientes de `v4x4`.

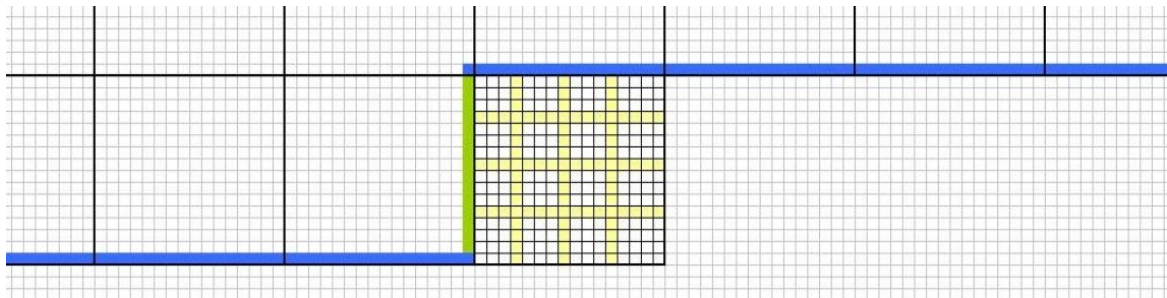


Figura 23: Amostras de luminância armazenadas em mem\_viz (blocos azuis e verdes) e em v4x4 (blocos amarelos) [4]

Como pode ser visto na figura 21, as quatro amostras geradas em cada ciclo são sempre horizontais. Portanto, para viabilizar o armazenamento de amostras verticais, utilizou-se o registrador **ord0**. Assim, o multiplexador **mux0** disponibiliza amostras horizontais ou verticais para as memórias mem\_viz e v4x4.

A tabela 5 mostra como é organizada a memória mem\_viz, a qual contém 1.024 conjuntos de 4 amostras. Como cada amostra é definida em 8 bits, tem-se 32 bits em cada endereço da memória.

**Tabela 5: Organização da memória mem\_viz**

Posição	Nº de amostras	Descrição
0 - 479	1920	Vizinhos superiores de luminância (blocos azuis na figura 23).
504 - 507	16	Vizinhos à esquerda de luminância (blocos verdes na figura 23).
508 - 509	8	Vizinhos à esquerda de croma Cb.
510 - 511	8	Vizinhos à esquerda de croma Cr.
512 - 751	960	Vizinhos superiores de croma Cb.
768 - 1007	960	Vizinhos à esquerda de croma Cr.

Já para a memória v4x4, verificou-se que 7 grupos de 4 amostras, totalizando 28 registradores, são suficientes para o armazenamento das vizinhanças dos blocos 4x4. A figura 24 mostra a denominação dada às vizinhanças 4x4, bem como as amostras que devem estar armazenadas na memória após a predição de cada bloco. Observa-se que somente após a

predição do bloco 10 todos os endereços de v4x4 são necessários. Já a figura 25 mostra quais as vizinhanças 4x4 estão armazenadas durante a predição de cada bloco.

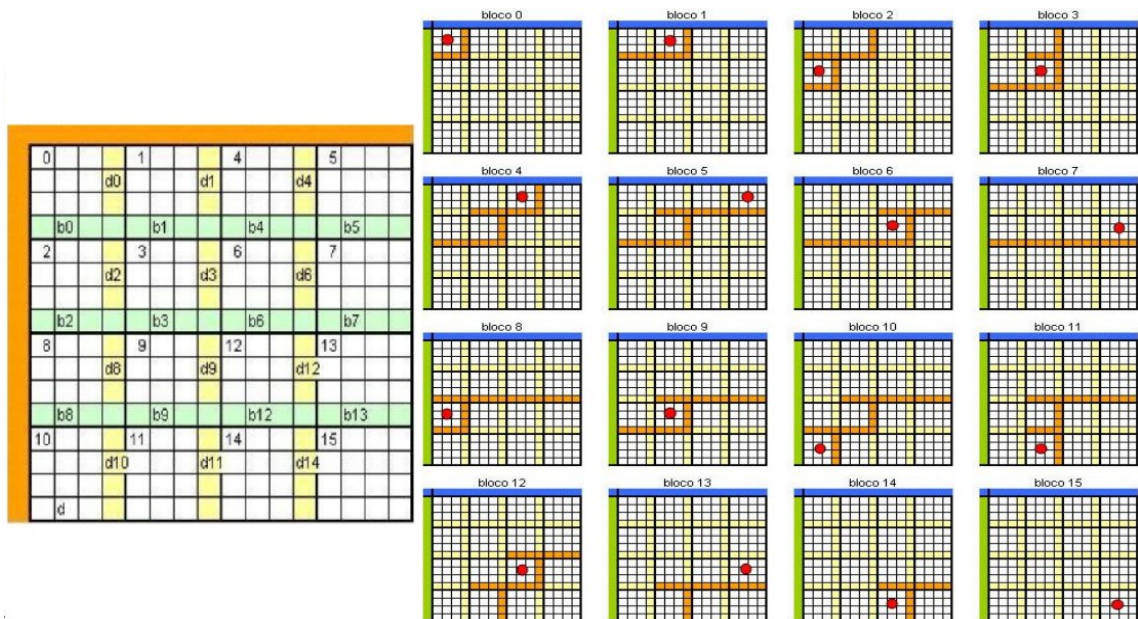


Figura 24: Denominação das vizinhanças 4x4, à esquerda.

Amostras que estão armazenadas em v4x4 após a predição de cada bloco, à direita [4]

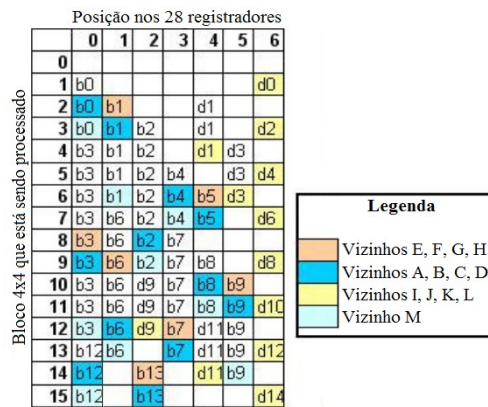


Figura 25: Grupos de amostras que estão armazenados em v4x4 durante a predição de cada bloco [4]

## 4. OTIMIZAÇÕES NO MÓDULO DE PREDIÇÃO INTRA-QUADROS

Duas otimizações foram consideradas ao longo deste trabalho: aumento da frequência de funcionamento do módulo *intra* e integração deste no decodificador de vídeo.

### 4.1 Aumento da frequência de funcionamento do módulo *intra*

Uma das maiores vantagens do sistema digital em relação ao analógico está na resolução da imagem. Enquanto o sistema PAL-M – sistema analógico utilizado no Brasil – possui uma resolução de 525 linhas horizontais, o SBTVD permite a transmissão de vídeo com até 1080 linhas horizontais. Maior resolução significa que cada quadro terá uma definição melhor e uma quantidade maior de informação.

Para que esta característica seja observada, o decodificador de vídeo deve ser capaz de funcionar, pelo menos, na frequência mínima para decodificação de vídeo em Full HD. Sabendo que o SBTVD utiliza sub-amostragem 4:2:0 (1,5 amostras por pixel) e uma taxa de 30 quadros por segundo, tem-se uma taxa de 93.312.000 amostras/segundo, como mostra a tabela 6.

**Tabela 6: Requisitos mínimos para o funcionamento do decodificador de vídeo**

<b>Taxa, em Amostras/s, necessária para vídeo em Full HD</b>	<b>Frequência mínima de operação do Intra considerando uma saída de 4 amostras/ciclo</b>	<b>Frequência de operação do decodificador de vídeo para HD</b>	<b>Frequência de operação do decodificador de vídeo para Full HD</b>
1920*1080*30*1,5 = 93312000	93312000/4 = 23,328 MHz	(Simulação) 50 MHz	(Simulação) 113,333 MHz

Considerando o funcionamento ideal do módulo de predição intra-quadros, teríamos, a cada ciclo, uma saída de 4 amostras. Conseqüentemente, uma frequência de 23,328 MHz seria suficiente para a geração de predições para vídeo em Full HD. Entretanto, o funcionamento



contínuo do bloco *intra* depende de outros blocos, como mostra a figura 26. Assim, para que isto fosse possível, as memórias FIFO que contêm as informações de resíduo e de contexto (*FIFO Resíduos* e *FIFO Contextos*) deveriam, sempre que solicitadas, disponibilizar os dados necessários. Já a memória de saída (*FIFO Video Out*) deveria ter sempre espaço disponível para o armazenamento de novas amostras.

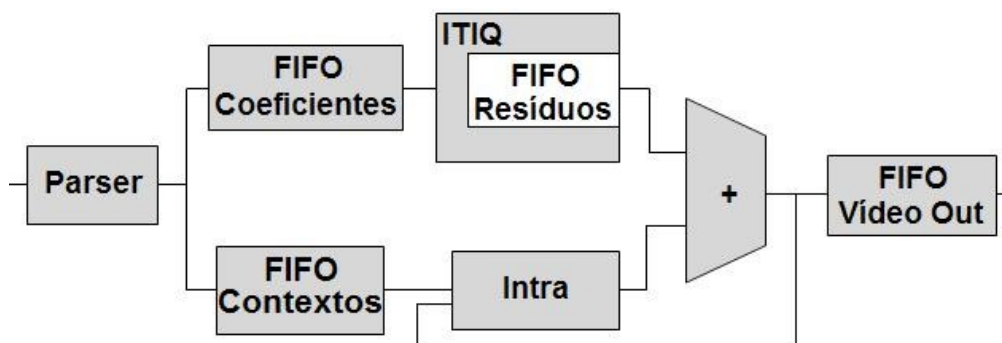


Figura 26: Blocos que influenciam no funcionamento do *intra*

Portanto, a frequência mínima de funcionamento do decodificador de vídeo foi determinada através de simulações. Ficou constatado que, para a decodificação de vídeo em HD, esta frequência é de 50 MHz. Já para a decodificação de vídeo em Full HD, a frequência deve ser superior a 113,333 MHz.

Como a frequência máxima do bloco de predição intra-quadros, implementado no FPGA Virtex 5 LX110T, ficou limitada à 74,485 MHz, a decodificação de vídeo em Full HD não era possível. Desta forma, otimizações foram realizadas para que esta frequência superasse os 113,333 MHz.

#### 4.1.1 Análise do caminho crítico

Em sistemas digitais, o caminho crítico é aquele que causa o maior atraso na propagação de sinais, limitando a frequência do circuito. Quanto maior for a lógica

combinatória entre elementos memorizantes, maior será o tempo necessário para que todos os sinais se estabilizem. Consequentemente, menor será a frequência máxima de relógio.

Desta forma, quando um sistema está limitado em frequência, uma das soluções é dividir o caminho crítico através da inserção de elementos memorizantes, como registradores. Como consequência, tem-se o aumento do número de ciclos de máquina necessários para o sinal chegar ao seu destino. Assim, o melhor resultado, em termos de performance, é aquele que minimiza a latência do circuito, isto é, o produto entre o número de ciclos e o período de cada ciclo.

A análise do caminho crítico no módulo de predição intra-quadros foi realizada com auxílio da ferramenta de síntese ISE, da *Xilinx*. Esta localiza e indica o tempo de atraso de cada elemento combinacional pertencente ao caminho crítico. Assim, constatou-se que os maiores atrasos ocorrem entre a entrada do módulo gerador de predições (GP) e a saída do somador que realimenta o módulo *intra*. A figura 27 mostra os elementos que fazem parte do caminho crítico. A localização deste, no decodificador de vídeo, é destacada na figura 28.

```
Data Path: intra_i/intra0/bctx1/saida_33 to intra_i/intra0/mv/mem0/Mram_mem_array2
```

Cell:in->out	fanout	Gate	Net	Delay	Delay	Logical Name (Net Name)
FDE: C->Q	1	0.396	0.850	intra_i/intra0/bctx1/saida_33	(intra_i/intra0/bctx1/saida_33)	
LUT6: I0->0	1	0.086	0.768	intra_i/intra0/G_P/Gera4x4/modo<1>92_SW0	(H1415)	
LUT5: I0->0	4	0.086	0.372	intra_i/intra0/G_P/Gera4x4/modo<1>92	(intra_i/intra0/G_P/Gera4x4/modo<1>92)	
LUT4: I3->0	62	0.086	0.642	intra_i/intra0/G_P/Gera4x4/modo<1>104	(intra_i/intra0/G_P/Gera4x4/modo<1>)	
LUT5: I2->0	31	0.086	0.450	intra_i/intra0/G_P/Gera4x4/predicao4/pc/se1Mux_PE1_3<1>1	(intra_i/intra0/G_P/Gera4x4/predicao4/w7<1>)	
LUT3: I2->0	1	0.086	0.611	intra_i/intra0/G_P/Gera4x4/predicao4/po/mux_PE1_3/Mux_dat_o_71	(intra_i/intra0/G_P/Gera4x4/predicao4/po/mux_PE1_3/Mux_dat_o_71)	
LUT6: I2->0	1	0.086	0.361	intra_i/intra0/G_P/Gera4x4/predicao4/po/se1Mux_PE1_3<3>1	(intra_i/intra0/G_P/Gera4x4/predicao4/po/viz3_PE1<0>)	
LUT6: I5->0	1	0.086	0.000	intra_i/intra0/G_P/Gera4x4/predicao4/po/pe1/Madd_temp2_addsub0001_lut<0>	(intra_i/intra0/G_P/Gera4x4/predicao4/po/pe1/Madd_temp2_addsub0001_lut<0>)	
MUXCY: S->0	1	0.305	0.000	intra_i/intra0/G_P/Gera4x4/predicao4/po/pe1/Madd_temp2_addsub0001_cyc<0>	(intra_i/intra0/G_P/Gera4x4/predicao4/po/pe1/Madd_temp2_addsub0001_cyc<0>)	
XORCY: CI->0	3	0.300	0.444	intra_i/intra0/G_P/Gera4x4/predicao4/po/pe1/Madd_temp2_addsub0001_xor<1>	(intra_i/intra0/G_P/Gera4x4/predicao4/po/pe1/temp2_addsub0001<1>)	
LUT6: I4->0	1	0.086	0.000	intra_i/intra0/G_P/Gera4x4/predicao4/po/pe1/Madd_temp2_Madd_lut<2>	(intra_i/intra0/G_P/Gera4x4/predicao4/po/pe1/Madd_temp2_Madd_lut<2>)	
MUXCY: S->0	1	0.305	0.000	intra_i/intra0/G_P/Gera4x4/predicao4/po/pe1/Madd_temp2_Madd_cy<2>	(intra_i/intra0/G_P/Gera4x4/predicao4/po/pe1/Madd_temp2_Madd_cy<2>)	
XORCY: CI->0	12	0.300	0.403	intra_i/intra0/G_P/Gera4x4/predicao4/po/pe1/Madd_temp2_Madd_xor<3>	(intra_i/intra0/G_P/Gera4x4/predicao4/po/pe1/temp2<3>)	
LUT6: I5->0	2	0.086	0.440	intra_i/intra0/G_P/Gera4x4/predicao4/po/pe1/saida<2>1_SW1	(H1720)	
LUT5: I3->0	18	0.086	0.915	intra_i/intra0/G_P/Gera4x4/predicao4/po/se1Mux_PE3_2<3>21	(intra_i/intra0/G_P/Gera4x4/predicao4/po/viz2_PE3<2>)	
LUT6: I0->0	1	0.086	0.000	intra_i/intra0/G_P/Gera4x4/predicao4/po/pe3/temp1<5>_SW4_G	(H1176)	
MUXF7: I1->0	2	0.214	0.365	intra_i/intra0/G_P/Gera4x4/predicao4/po/pe3/temp1<5>_SW4	(H1401)	
LUT6: I5->0	2	0.086	0.365	intra_i/intra0/G_P/Gera4x4/predicao4/po/pe3/temp1<5>	(intra_i/intra0/G_P/Gera4x4/predicao4/po/pe3/temp1<5>)	
LUT6: I5->0	1	0.086	0.000	intra_i/intra0/G_P/Gera4x4/predicao4/po/pe3/Madd_temp2_Madd_lut<5>	(intra_i/intra0/G_P/Gera4x4/predicao4/po/pe3/Madd_temp2_Madd_lut<5>)	
MUXCY: S->0	1	0.305	0.000	intra_i/intra0/G_P/Gera4x4/predicao4/po/pe3/Madd_temp2_Madd_cy<5>	(intra_i/intra0/G_P/Gera4x4/predicao4/po/pe3/Madd_temp2_Madd_cy<5>)	
XORCY: CI->0	6	0.300	0.380	intra_i/intra0/G_P/Gera4x4/predicao4/po/pe3/Madd_temp2_Madd_xor<6>	(intra_i/intra0/G_P/Gera4x4/predicao4/po/pe3/temp2<6>)	
LUT6: I5->0	1	0.086	0.000	sum_sel_i/pred3<6>68_SW0_G	(H1095)	
MUXF7: I1->0	1	0.214	0.436	sum_sel_i/pred3<6>68_SW0	(H980)	
LUT6: I4->0	1	0.086	0.000	sum_sel_i/Madd_sum0_lut<6>	(sum_sel_i/Madd_sum0_lut<6>)	
MUXCY: S->0	1	0.305	0.000	sum_sel_i/Madd_sum0_cy<6>	(sum_sel_i/Madd_sum0_cy<6>)	
MUXCY: CI->0	0	0.023	0.000	sum_sel_i/Madd_sum0_cy<7>	(sum_sel_i/Madd_sum0_cy<7>)	
XORCY: CI->0	16	0.300	0.419	sum_sel_i/Madd_sum0_xor<8>	(sum_sel_i/Madd_sum0_xor<8>)	
LUT4: I3->0	8	0.086	0.262	intra_i/intra0/mux0/saida_3_mux0000<4>1	(intra_i/intra0/fio_mem_ent<3><4>)	
RAMB18: DIAL		0.299		intra_i/intra0/mv/mem0/Mram_mem_array2		

```
Total 13.426ns (4.942ns logic, 8.484ns route)
(36.8% logic, 63.2% route)
```

Figura 27: Caminho crítico do módulo de predição intra-quadros

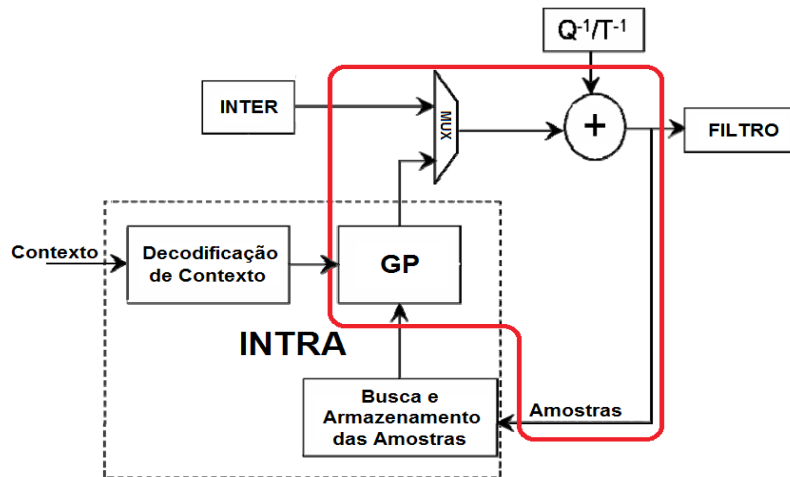


Figura 28: Localização do caminho crítico no decodificador de vídeo

O atraso total provocado por estes elementos é de 13,426 ns, limitando a frequência do sistema em 74,485 MHz.

#### 4.1.2 Inserção de registradores no caminho crítico

Conhecendo o atraso de cada elemento combinacional, a inserção de registradores é feita o mais próximo possível do centro do caminho crítico, o que reduziria o atraso total neste pela metade. Isto não significa que a frequência máxima do sistema irá dobrar, visto que provavelmente o novo caminho crítico será outro.

No módulo *intra*, a inserção dos registradores foi realizada no interior dos blocos **PEs**, presentes no módulo GP4x4 (ver figura 20). Com isso, o atraso total do novo caminho crítico passou a ser de 8,048 ns, possibilitando uma frequência de até 124,252 MHz, suficiente para decodificação de vídeo em full HD.

Como consequência da inserção dos registradores, uma completa re-engenharia do módulo de controle do *intra* foi necessária para manter o seu funcionamento correto.

### 4.1.3 Re-engenharia do módulo de controle

A principal consequência da inserção de registradores no módulo gerador de predições é a necessidade de um ciclo exclusivamente para o armazenamento das amostras. Antes, as amostras geradas no ciclo atual poderiam ser utilizadas como vizinhanças para a geração das amostras do ciclo seguinte. Já com a inserção dos registradores, houve a necessidade de adicionar um ciclo para o armazenamento dessas amostras que serão utilizadas no ciclo seguinte. A figura 29 mostra o diagrama de tempos para a memorização das amostras na versão do módulo *intra* sem registradores. Já a figura 30 mostra o mesmo diagrama para a nova versão.

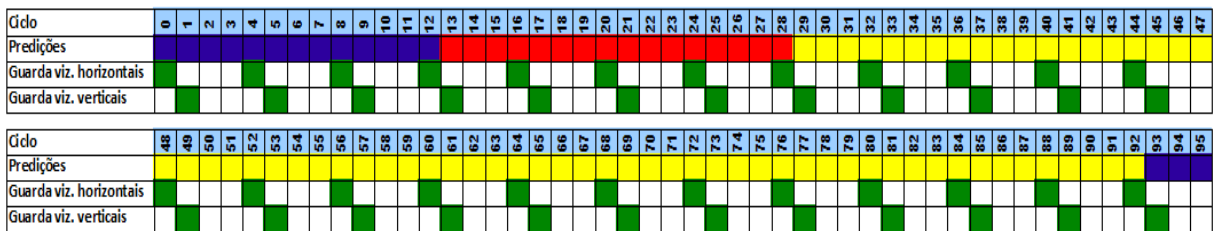


Figura 29: Armazenamento das vizinhanças na versão do *intra* sem registradores

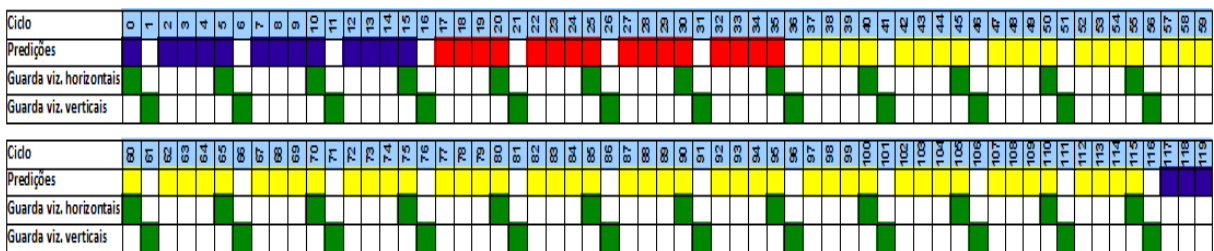


Figura 30: Armazenamento das vizinhanças na versão do *intra* com registradores

Como pode-se observar na figura 30, após a geração das últimas quatro amostras de cada bloco, um ciclo foi adicionado exclusivamente para o armazenamento destas. Todos os outros blocos do *intra* são desabilitados e a saída é desconsiderada durante estes ciclos. Assim, apenas os módulos **mem\_viz** e **v4x4** são atualizados. Ao final destes ciclos, as

entradas dos novos registradores estarão com os sinais adequados para a geração das previsões do ciclo seguinte.

A figura 31 mostra o comportamento de alguns sinais de controle do módulo *intra*, durante a previsão de um bloco 4x4 de luminâncias.

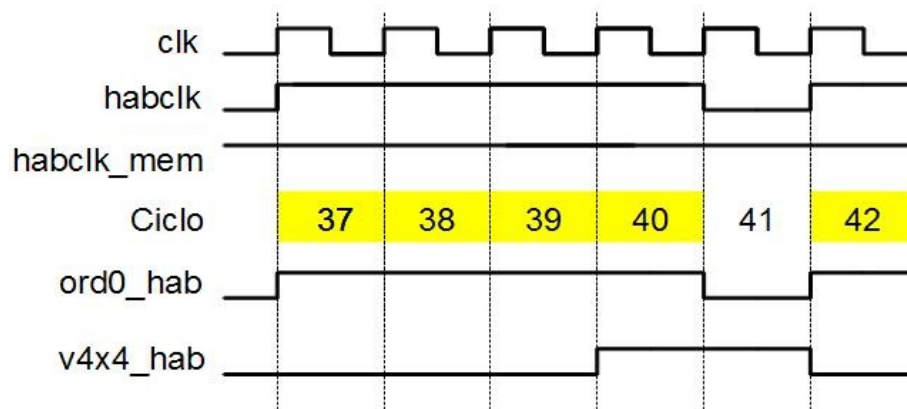


Figura 31: Alguns sinais de controle do módulo *intra* (situação 1)

Nos ciclos 37 à 40, as 16 amostras de luminância do bloco 0 são geradas (4 amostras por ciclo). O sinal **ord0\_hab** habilita o armazenamento da quarta amostra de cada ciclo em *ord0*. Estas amostras fazem parte da vizinhança esquerda do bloco 1. No ciclo 40, a memória *v4x4* é habilitada para guardar as 4 amostras geradas neste ciclo, as quais fazem parte da vizinhança superior do bloco 2. Já no ciclo 41, as amostras verticais, armazenadas em *ord0*, são memorizadas em *v4x4*.

O sinal **habclk** habilita todos os elementos memorizantes do módulo *intra*, com exceção de *mem\_viz* e *v4x4*, os quais são habilitados por **habclk\_mem**. A figura 32 mostra o comportamento destes sinais quando a FIFO de resíduos está vazia. O mesmo comportamento é observado quando a FIFO de contextos está vazia ou quando a FIFO de saída está completa (ver figura 26).

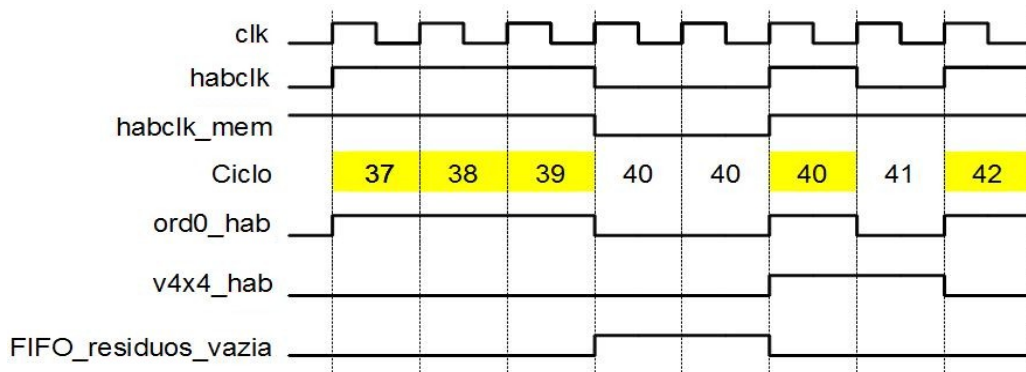


Figura 32: Alguns sinais de controle do módulo *intra* (situação 2)

Como pode-se observar, o funcionamento do módulo *intra* é interrompido enquanto não há resíduos disponíveis.

Além dos registradores adicionados no caminho crítico, outros registradores também foram adicionados para sincronizar os sinais de controle com as saídas do *intra*. Desta forma, o caminho crítico final provoca um atraso de apenas 7,568 ns, possibilitando uma frequência de até 132,129 MHz.

Como a cada 5 ciclos o módulo *intra* está desabilitado em um, a arquitetura é capaz de gerar amostras em 80% dos ciclos, sendo equivalente a uma frequência máxima de funcionamento de 105,7 MHz. Entretanto, é importante salientar que, para garantir o funcionamento da decodificação de vídeo em full HD, a frequência máxima de síntese do decodificador é considerada. Desta forma, os 132,129 MHz são suficientes.

## 4.2 Integração do módulo *intra* no decodificador de vídeo

A integração do módulo *intra* no decodificador de vídeo, juntamente com o módulo *inter* (também conhecido como módulo de compensação de movimento ou *MC*), é fundamental para a decodificação de quadros **I**, **P** e **B**.

Quadro **I** é aquele que possui apenas predição intra-quadros.

Quadro **P** possui predição inter-quadros com referência em quadros passados e pode conter macroblocos com predição intra-quadros.

Quadro **B** possui predição inter-quadros com referência em quadros passados e quadros futuros, isto é, quadros que já foram decodificados, mas ainda não foram exibidos. Também pode conter macroblocos com predição intra-quadros.

Desta forma, o módulo *intra* isolado permite apenas a decodificação de quadros I. Já o módulo *MC* isolado permite a decodificação de quadros P e B incompletos, sem a presença de macroblocos com predição intra-quadros.

Para viabilizar a integração destes módulos no decodificador de vídeo, quatro pontos devem ser considerados:

- Identificação do tipo de predição de cada macrobloco: intra-quadros ou inter-quadros;
- Armazenamento das amostras geradas pelo *MC* na memória do *intra*;
- Garantir o sincronismo na geração das predições;
- Evitar situações de *deadlock* (travamento), nas quais o sistema fica bloqueado em um estado.

#### **4.2.1 Identificação do tipo de predição de cada macrobloco**

O primeiro passo para a integração dos módulos *intra* e *MC* foi a identificação do tipo de predição de cada macrobloco: intra-quadros ou inter-quadros. Para isso, foi necessário compreender o funcionamento de outro módulo presente no decodificador de vídeo: o *parser* (ver figura 26). Este é responsável pela entrega dos sinais relativos ao *contexto* de cada macrobloco, além de controlar a entrega dos coeficientes de resíduos (transformados e quantizados), através de uma FIFO, ao bloco ITIQ, o qual realiza a transformada inversa e a quantização inversa destes. A arquitetura do *parser* é apresentada na figura 33.

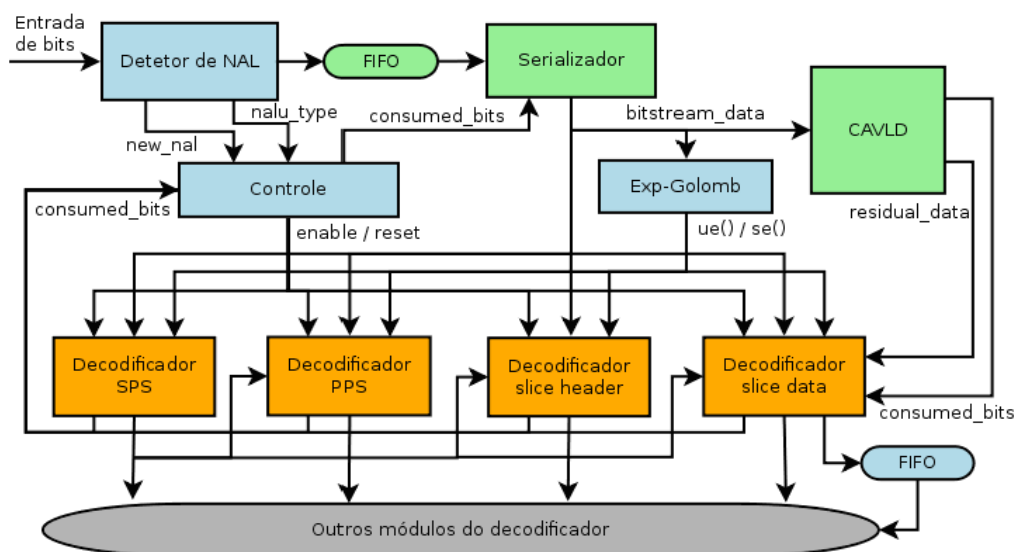


Figura 33: Arquitetura do módulo *parser* [3]

Analisando esta arquitetura, constatou-se que o sinal *mb\_type*, presente no módulo *Decodificador slice data*, poderia ser utilizado para a identificação do tipo de predição de cada macrobloco. Quando este sinal é superior a 25, tem-se predição inter-quadros. Caso contrário, tem-se predição intra-quadros.

Desta forma, o sinal *MB\_TP* foi criado, no bloco *slice data*, para realizar esta identificação. Este sinal é repassado à FIFO de contextos, de forma que os módulos de predição e o somador recebam esta informação no momento certo. Assim, quando há uma leitura desta FIFO, identifica-se o tipo de predição do macrobloco atual, de forma que o bloco somador adicione o resíduo à saída do módulo de predição correspondente.

#### 4.2.2 Armazenamento das amostras geradas pelo *MC* na memória do *intra*

Antes da integração dos módulos de predição, as predições geradas no *intra* eram somadas ao resíduo, através do somador, e as amostras resultantes eram armazenadas nas memórias *mem\_viz* e *v4x4*. Para armazenar as amostras geradas pelo *MC* nestas memórias, foi



necessário acrescentar um multiplexador entre os módulos de predição e o somador, como é mostrado na figura 34, e garantir que o módulo *intra* continue em funcionamento durante a geração das predições inter-quadros.

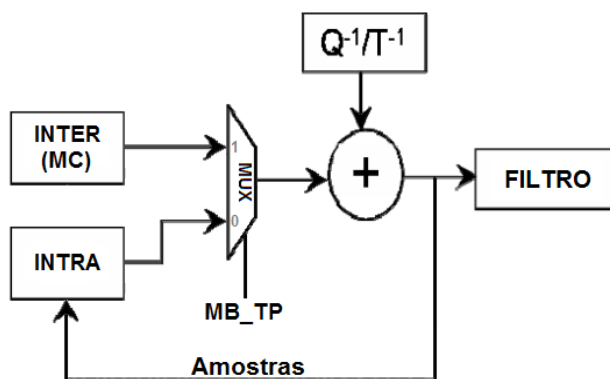


Figura 34: Multiplexação dos módulos de predição através de MB\_TP

Para a decodificação de quadros P e B, não foi necessário refazer o bloco de decodificação de contexto do *intra*. Ao invés disto, o *parser* foi modificado de forma que o módulo de predição intra-quadros receba um *contexto* pré-determinado quando o macrobloco tiver predição inter-quadros.

Além disso, quando ocorria uma transição de um quadro I para um quadro P, as predições intra-quadros das componentes de crominância do último macrobloco do quadro I não eram concluídas, devido ao travamento do módulo *intra* por falta de *contexto*.

As modificações no *parser* que solucionaram os problemas citados acima também envolveram o módulo *slice data*. Quando o sinal *mb\_type* é atualizado, e identifica-se predição inter-quadros, tem-se a escrita do *contexto* pré-determinado na FIFO de contextos. A única informação relevante deste *contexto* é a indicação de que a predição do macrobloco atual é realizada pelo módulo *MC*.

Assim, evita-se o travamento do *intra* por falta de *contexto* na transição de um macrobloco com predição intra-quadros para um macrobloco com predição inter-quadros, e

possibilita-se o armazenamento das amostras geradas pelo módulo *MC*, somadas ao resíduo, nas memórias de vizinhanças do *intra*. Neste caso, as amostras geradas pelo módulo *intra* são desconsideradas.

### **4.2.3 Sincronização das amostras geradas**

Garantir o sincronismo na geração das predições é necessário para que as informações de resíduo sejam somadas às amostras corretas. Desta forma, quando o funcionamento de um módulo de predição é interrompido, o outro também deve parar. Por este motivo, o sinal *intra\_ok\_w* sinaliza quando a saída dos módulos de predição é válida.

Como foi visto no subcapítulo anterior, quando um macrobloco com predição inter-quadros está sendo decodificado, o módulo de predição intra-quadros continua em funcionamento e tem suas predições desconsideradas. Assim, as predições geradas pelo *MC* são somadas ao resíduo e obtém-se as amostras decodificadas. Quando a saída do módulo *MC* não é válida, o módulo *intra* é desabilitado para não perder o sincronismo na geração das predições e soma destas aos resíduos correspondentes.

Durante a decodificação de um macrobloco com predição intra-quadros, o módulo *MC* continua, se possível, gerando as predições e armazenando-as em uma FIFO, enquanto esta não estiver completa. A leitura desta FIFO só é realizada quando o macrobloco a ser decodificado tiver predição inter-quadros, passando a ocorrer o que foi descrito no parágrafo anterior.

### **4.2.4 Situações de *deadlock***

Durante a integração dos módulos de predição, constatou-se que estes permaneciam bloqueados em alguns estados, devido a existência de macroblocos do tipo SKIP. Estes

ocorrem quando o vetor de movimento e o resíduo associado possuírem apenas valores nulos. Assim, quando o codificador sinaliza que um macrobloco é do tipo SKIP, este será a cópia do macrobloco de mesma posição do quadro de referência anterior, não havendo a necessidade do envio de resíduos.

Como o funcionamento do *intra* é interrompido na ausência de resíduos na FIFO, o *parser* continua em funcionamento até que a FIFO de contextos não tenha mais endereços disponíveis. Neste caso, tem-se uma situação de *deadlock*: o *intra* não faz nenhuma leitura na FIFO de contextos devido à ausência de resíduos na FIFO de resíduos, e o *parser* não escreve novos coeficientes de resíduos na FIFO de coeficientes porque a FIFO de contextos está completa, provocando seu travamento.

Para corrigir esta situação, o *parser* foi forçado a gerar coeficientes de resíduos nulos na presença de macroblocos do tipo SKIP. Assim que ocorre a identificação de um macrobloco deste tipo no módulo *slice data*, 96 escritas de coeficientes nulos são realizadas na FIFO de coeficientes, uma para cada LOP (conjunto de 4 amostras). Além disso, um *contexto* pré-determinado é escrito na FIFO de contextos, como ocorre em macroblocos com predição inter-quadros.

Além destas quatro considerações, algumas modificações foram realizadas no *parser* para corrigir o fornecimento de coeficientes à FIFO de coeficientes, de acordo com os erros observados em simulações.

## 5. RESULTADOS

A validação do sistema é feita através de simulações, nas quais um *bitstream* (sequência de bits) conhecido é aplicado na entrada, e as saídas obtidas são comparadas aos valores corretos. As simulações são realizadas através da ferramenta *ModelSim*, a qual analisa um arquivo de teste (*testbench*) e gera as formas de onda dos sinais selecionados.

Para a validação da integração dos módulos de predição, as saídas do *MC*, do *intra* e do somador são comparadas aos valores corretos, os quais são gerados através da ferramenta *PRH.264*, a qual foi desenvolvida e validada pelo LaPSI.

Quanto às modificações do módulo *intra* visando o aumento da frequência máxima de funcionamento, os resultados são sintetizados nas tabelas 7 e 8.

**Tabela 7: Desempenho das versões anterior e otimizada do *intra***

DESEMPENHO – Virtex 5 XC5VLX110T		
	Versão Anterior	Versão Otimizada
Período mínimo	13.426 ns	7.568 ns
Frequência máxima	74.485 MHz	132.129 MHz

**Tabela 8: Recursos do FPGA utilizados nas versões anterior e otimizada do *intra***

ÁREA – Virtex 5 XC5VLX110T	
Versão Anterior	Recursos consumidos (% do total)
Número de Registradores	2018 (2%)
Número de LUTs	4355 (6%)
Número de BRAMs	2 (1%)

ÁREA – Virtex 5 XC5VLX110T	
Versão Modificada	Recursos consumidos (% do total)
Número de Registradores	2110 (3%)
Número de LUTs	4428 (6%)
Número de BRAMs	2 (1%)

Como pode-se observar, houve uma melhora de 77,4% em termos de desempenho para um acréscimo de somente 4,56% de registradores e 1,68% de LUTs (elementos combinacionais do FPGA). O número de *Block RAMs* (BRAMs) não foi alterado.

## 6. CONCLUSÕES

Otimizar um módulo com a complexidade do *intra* requer a máxima compreensão de seu funcionamento e de sua implementação. Uma pequena mudança em sua descrição em VHDL pode ocasionar erros que precisam de algumas horas – ou de alguns dias – de depuração para corrigi-los.

A otimização do módulo de predição intra-quadros foi realizada com êxito. Através da inserção de registradores no caminho crítico e da re-engenharia do módulo de controle, houve um aumento de 77,4% na frequência máxima de funcionamento do *intra*, alcançando uma frequência de 132,129 MHz, valor superior aos 113,333 MHz necessários para a decodificação de vídeo em full HD.

O trabalho de integração do *intra* no decodificador de vídeo possibilita a decodificação de *bitstreams* com quadros I, P e B. Para isso, modificações foram realizadas no módulo *parser*, possibilitando a identificação do tipo de predição de cada macrobloco, bem como o armazenamento de amostras geradas através de predição inter-quadros nas memórias de vizinhanças do módulo de predição intra-quadros. Além disso, coeficientes nulos são armazenados na FIFO de coeficientes na presença de um macrobloco do tipo SKIP, evitando situações de *deadlock*.

Desta forma, este projeto contemplou diversos aspectos no campo de sistemas digitais: análise de caminho crítico, otimização de desempenho, controle e integração de diversos módulos, utilização de FIFOs, criação de *testbenches*, simulações comportamentais, depuração de erros, entre outros.

A otimização do *intra*, visando a geração de predições intra-quadros para sequências de vídeo codificadas através do modo entrelaçado, pode ser tema de trabalhos futuros.

## 7. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] LULA DA SILVA, L. I. Atos do Poder Executivo. **Decreto N° 4.901**. Edição Número 231 de 27 de nov. de 2003. Publicado no Diário Oficial da União. Disponível em: <[http://sbtvd.epqd.com.br/downloads/decreto\\_4901\\_2003.pdf](http://sbtvd.epqd.com.br/downloads/decreto_4901_2003.pdf)>. Acesso em: 7 abr. 2011.
- [2] RICHARDSON, I. E. G. **H.264 and MPEG-4 Video Compression - Video Coding for Next-Generation Multimedia**. John Wiley and Sons, Chichester, 2003. ISBN: 0-470-84837-5.
- [3] LORENCETTI, M. A. **Parser em VHDL para decodificador de vídeo H.264 para SBTVD**. 51 p. Projeto de Diplomação em Engenharia Elétrica – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2010.
- [4] BERRIEL, E. A. **O Padrão de Codificação de Vídeo H.264 e uma Implementação do Módulo de Predição Intra Quadro em Hardware**. 71 p. Programa de Pós-Graduação em Computação – Universidade Federal do Rio Grande do Sul, Porto Alegre, Jul. 2005.
- [5] International Telecommunication Union. **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. Nov. 2007.

- [6] AGOSTINI, Luciano V.; AZEVEDO FILHO, Arnaldo P.; STAEBLER, Wagston T.; ROSA, Vagner S.; ZATT, Bruno; PINTO, Ana Cristina M.; PORTO, Roger Endrigo; BAMPI, Sergio; SUSIN, Altamiro A. **Design and FPGA Prototyping of a H.264/AVC Main Profile Decoder for HDTV**. Journal of the Brazilian Computer Society, v. 13, p. 25-36, 2007.
- [7] LaPSI, **Laboratório de Processamento de Sinais e Imagens**.  
<<http://www.lapsi.eletr.ufrgs.br/>>. Acesso em: 25 jun. 2011.
- [8] DE SOUZA, M. F. **O sistema terrestre mais avançado do mundo**. Jul. 2010.  
<<http://www.observatoriodaimprensa.com.br/news/view/o-sistema-terrestre-mais-avancado-do-mundo>>. Acesso em: 25 jun. 2011.