Dissertação de Mestrado

Inatel
Instituto Nacional de Telecomunicações

UMA PROPOSTA DE IMPLEMENTAÇÃO DO MULTIPLEXADOR DO SISTEMA ISDB-T_B

JULIANO SILVEIRA FERREIRA

OUTUBRO/ 2009

Uma proposta de implementação do Multiplexador do sistema ISDB-T_B

Juliano Silveira Ferreira

Dissertação apresentada ao Instituto Nacional de Telecomunições, como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Geraldo Gil Ramundo Gomes

Co-orientador: Prof. Dr. Luciano Leonel Mendes

Santa Rita do Sapucaí, MG
2009

Dissertação defendida e aprovada em 22/10/2009, pela comissão julgadora:

Prof. Dr. Geraldo Gil Ramundo Gomes (Orientador) - Inatel

Prof. Dr. Yuzo Iano - Unicamp

Prof. Dr. Carlos Alberto Ynoguti - Inatel

Prof. Dr. José Câmara Brito

Primeiramente a Deus. A
meus pais, irmãos e à
minha esposa Hannagely e
meu filho Guilherme
agradeço por todo carinho,
apoio e paciência. A todos
os amigos e companheiros
de trabalho cuja
colaboração foi essencial
para esta conquista.

Agradecimentos

Primeiramente agradeço a Deus por se fazer presente em minha vida e por todo carinho e cuidado demonstrado com relação a minha pessoa e família.

Agradeço a orientação, colaboração e incentivo de meus orientadores Prof Dr. Geraldo Gil Ramundo Gomes e Prof Dr. Luciano Leonel Mendes.

Agradeço a meus pais José e Maria de Lourdes e a meu avô Joaquim a oportunidade de estudar e por serem minha referência sólida de caráter e honestidade. Apesar de todas as dificuldades e sacrifícios vocês nunca deixaram de me apoiar e incentivar, assim como meus irmãos Jaqueline, José Henrique e Luciana.

Agradeço a paciência, carinho e compreensão de minha esposa Hannagely e de meu filho Guilherme não só durante o processo de desenvolvimento deste trabalho, mas em todos os momentos de nossa caminhada juntos.

Agradecimento a Carlos Augusto Rocha pela amizade e companheirismo e por acreditar em meu potencial e me proporcionar inúmeras oportunidades de crescimento profissional e pessoal. A Marcelo Carneiro de Paiva e Antônio Alves Ferreira Jr. o agradecimento pelo incentivo e força para concluir este trabalho. Meus sinceros agradecimentos aos que contribuíram de maneira ativa e essencial para toda pesquisa e desenvolvimento envolvidos durante esta caminhada: Bruno Borsato de Souza Lima, Caio Pereira Torcato, Cléomarcio Almeida, Danilo Simões Gaspar, Dário Daniel Ribeiro de Morais, Farley Igor Martins Balbino, Ivan Simões Gaspar e Rafael Adami Pivoto.

Agradecimento ao Inatel e a empresa Linear Equipamentos Eletrônicos S/A pela oportunidade de poder trabalhar em suas dependências, usufruir de sua infraestrutura e principalmente por poder interagir com as pessoas que forma tais instituições que tanto colaboram para meu desenvolvimento profissional e pessoal.

Enfim, agradeço a todos que apoiaram e incentivaram diretamente ou indiretamente para que essa conquista fosse alcançada.

Índice

Lista de Figuras **vii**

Lista de Tabelas **x**

Lista de Abreviaturas e Siglas **xii**

Lista de Símbolos **xv**

1 Introdução **1**
1.1 Motivação 2
1.2 Contribuições 3
1.3 Estrutura da dissertação 4

2 Camada física do padrão ISDB-T$_B$ **5**
2.1 Sistema Brasileiro de TV digital 5
2.2 Sub-sistema de modulação 6
2.2.1 Re-multiplexador 8
2.2.2 Codificador Reed-Solomon 9
2.2.3 Aleatorizador 9
2.2.4 Ajuste de atraso 10
2.2.5 Entrelaçador de *bytes* 10
2.2.6 Codificador convolucional 11
2.2.7 Mapeamento das portadoras 13
2.2.8 Segmentos de dados 15
2.2.9 Combinador de níveis hierárquicos 16
2.2.10 Entrelaçador Temporal 17
2.2.11 Entrelaçador no domínio da frequência 18
2.2.12 Sinal OFDM 19
2.2.13 A capacidade de transmissão do sistema 26
2.3 Conclusão 28

3 Camada de multiplexação do padrão ISDB-T_B 30
3.1 Fluxo de dados da camada de transporte do padrão MPEG-2 . . . 30
3.1.1 Organização dos dados comprimidos . . . 31
3.1.2 Formação dos pacotes do TS . . . 31
3.1.3 Referências de tempo do TS . . . 35
3.1.4 Tabelas do sistema . . . 38
3.1.5 As tabelas SI/PSI . . . 38
3.1.6 Tabelas adicionais . . . 45
3.2 Multiplexação de TS MPEG-2 . . . 48
3.3 Geração do BTS . . . 48
3.3.1 Número de pacotes por quadro de multiplexação . . . 49
3.3.2 Organização dos pacotes no quadro de multiplexação . . . 50
3.3.3 *Bytes* adicionais nos pacotes do BTS . . . 51
3.3.4 IIP . . . 51
3.4 Conclusão . . . 54

4 Implementação e validação do multiplexador 59
4.1 Desafios da Implementação . . . 59
4.2 Modelo para implementação do sistema de multiplexação . . . 60
4.3 Técnicas e Ferramentas para Implementação do multiplexador . . . 62
4.3.1 *Hardware* utilizado . . . 62
4.3.2 Tecnologia FPGA . . . 62
4.3.3 Etapas de desenvolvimento de projetos em FPGA . . . 64
4.3.4 VHDL . . . 65
4.3.5 O ambiente de desenvolvimento . . . 66
4.3.6 Processador Nios II . . . 68
4.3.7 Sistema operacional *uClinux* . . . 69
4.3.8 O servidor WEB BOA . . . 70
4.4 Implementação do sistema de multiplexação . . . 71
4.4.1 Tratamento do TS de entrada . . . 71
4.4.2 *Buffer* de TS . . . 80
4.4.3 Configurações do sistema . . . 81
4.4.4 Multiplexador de TS . . . 82
4.4.5 Controle de leitura do MUX de TS . . . 84
4.4.6 Geração de tabelas do sistema . . . 84
4.4.7 Geração da IIP . . . 86
4.4.8 Geração dos *bytes* adicionais . . . 89
4.4.9 Tratamento do TS de saída . . . 93
4.5 Testes do protótipo . . . 94

4.5.1 Testes utilizando interfaces ASI da empresa Dektec e *softwares* de análise 94
4.5.2 Teste utilizando equipamentos profissionais de transmissão 103
4.6 Conclusão 109

5 Conclusão **115**

A Artigos publicados desta dissertação **117**

Bibliografia **118**

Lista de Figuras

2.1 Sub-sistema de modulação de TV digital. 6
2.2 Sub-sistema de transmissão e o multiplexador. 6
2.3 Diagrama em blocos do sub-sistema de modulação 7
2.4 O re-multiplexador e o codificador RS. 8
2.5 Pacote MPEG acrescido de 16 bytes do RS. 9
2.6 Circuito gerador da sequência pseudo-aleatória. 10
2.7 Estrutura do entrelaçador de bytes. 11
2.8 Diagrama em blocos do codificador convolucional com taxa 1/2. . 12
2.9 Processo de mapeamento. 13
2.10 Estrutura dos entrelaçadores de bits 14
2.11 Constelação QPSK. 14
2.12 Constelação 16QAM. 15
2.13 Constelação 64QAM. 15
2.14 O símbolo OFDM e os segmentos de dados 16
2.15 Exemplo de organização dos dados no combinador. 17
2.16 Estrutura do entrelaçador temporal. 17
2.17 Entrelaçador no domínio da frequência. 19
2.18 O intervalo de guarda. 20
2.19 Segmentação da banda e formação de cada segmento. 21
2.20 Exemplo de transmissão hierárquica: duas camadas ativas. 22
2.21 Exemplo de transmissão hierárquica: três camadas ativas. 22
2.22 Formação, em bits, do quadro da TMCC. 24

3.1 Formação do ES, PES e TS. 32
3.2 Formação do cabeçalho do TS. 32
3.3 Pacote do TS com adaptation field. 34
3.4 Pacote transportando o PCR. 36
3.5 Estrutura de um TS definida pelas tabelas PAT e PMT. 39
3.6 Estrutura de formação da PAT. 41
3.7 Estrutura de formação da PMT. 43
3.8 Estrutura de formação da CAT. 44

3.9 Estrutura de formação da NIT. 44
3.10 Multiplexação de dois TS's. 48
3.11 Exemplo de formação do quadro de multiplexação. 50
3.12 O receptor modelo do sistema ISDB. 51
3.13 Gerador, em MATLAB, do quadro de multiplexação. 52
3.14 Adição de 16 bytes ao pacote do TS. 52
3.15 Estrutura de formação do IIP. 54

4.1 Diagrama do modelo do multiplexador implementado. 61
4.2 Foto do *hardware* utilizado na implementação. 63
4.3 Estrutura básica de formação de um FPGA. 64
4.4 Visão do software Quartus II da Altera. 66
4.5 Visão da ferramenta SignalTap. 67
4.6 Visão da ferramenta Nios II IDE. 68
4.7 Configuração da CPU do Nios no software SOPC Builder. 70
4.8 Diagrama do multiplexador implementado. 71
4.9 Tratamento do TS de entrada. 72
4.10 A interface paralela de dados no SignalTap. 74
4.11 Primeira fase do tratamento do PCR. 75
4.12 Primeira etapa do tratamento de PCR vista no SignalTap. 76
4.13 O analisador de TS no SignalTap. 79
4.14 Organização dos dados extraídos da PAT. 80
4.15 Análise no StreamXpert da tabela PAT do fluxo de saída do multiplexador. 80
4.16 Diagrama em blocos do buffer de TS. 81
4.17 Alarmes e configuração da camada de transmissão de cada TS. . . 82
4.18 Tela de configuração dos parâmetros do transmissão. 83
4.19 Tela de configuração de data e hora do multiplexador. 83
4.20 Multiplexador de TS. 85
4.21 Sinais da geração de tabela e do multiplexador de TS. 87
4.22 Instante do envio da tabela PAT. 88
4.23 O envio da IIP no último pacote do quadro de multiplexação. . . 90
4.24 Instante de envio da IIP. 91
4.25 Inserção dos 16 bytes adicionais. 92
4.26 Segunda fase do tratamento do PCR. 93
4.27 Protótipo montado em um rack. 94
4.28 Protótipo em teste com as interfaces Dektec. 95
4.29 Software StremXpress enviando fluxo às interfaces ASI. 96
4.30 Análise no StreamXpert do BTS de saída do multiplexador. . . . 97
4.31 Análise de erros pelo StreamXpert segundo a ETR 290/TR101290. 99

4.32 Análise de PCR do fluxo LD no StreamXpert. 100
4.33 Análise de PCR do fluxo HD no StreamXpert. 100
4.34 Decodificação do fluxo LD no StreamXpert. 101
4.35 Decodificação do fluxo HD no StreamXpert. 101
4.36 Análise do BTS no MPEG TS *Compliance Analyzer*. 102
4.37 Verificação de erros do BTS pelo MPEG TS *Compliance Analyzer*. 103
4.38 Análise do PCR do programa LD pelo MPEG TS *Compliance Analyzer*. 104
4.39 Análise do PCR do programa HD pelo MPEG TS *Compliance Analyzer*. 105
4.40 Informações de data e hora extraídas da TOT. 105
4.41 Diagrama de transmissão utilizado no teste. 106
4.42 Equipamentos de transmissão utilizados no teste. 107
4.43 Análise do BTS de saída do multiplexador. 108
4.44 Análise de integridade do BTS de saída do multiplexador. 109
4.45 Análise de PCR do fluxo LD do BTS. 110
4.46 Análise de PCR do fluxo SD do BTS. 111
4.47 Análise de PCR do fluxo HD do BTS. 111
4.48 Decodificação do fluxo LD pelo StremXpert. 112
4.49 Decodificação do fluxo SD pelo StremXpert. 112
4.50 Decodificação do fluxo HD pelo StremXpert. 113
4.51 Os parâmetros de transmissão do modulador. 113
4.52 Decodificação do sinal LD no receptor móvel e no StreamXpert. . 113
4.53 Decodificação do sinal SD no receptor fixo. 114
4.54 Decodificação do sinal HD no receptor fixo. 114

Lista de Tabelas

2.1 Ajuste de atraso. 10
2.2 Puncionamento. 12
2.3 Ajuste de atraso em bits. 13
2.4 Número de portadoras de dados úteis de acordo com o modo. . . . 16
2.5 Possíveis valores de I e nc de acordo com o Modo. 18
2.6 Comprimentos do entrelaçador temporal e atrasos necessários. . . 19
2.7 Duração do símbolo e do quadro OFDM. 21
2.8 Exemplo de configuração utilizada nas transmissões iniciais no Brasil. 22
2.9 Números e tipos de portadoras do padrão ISDB-T_B. 23
2.10 TMCC - Os 102 bits de informações do sistema. 25
2.11 Descrição da configuração de cada camada. 25
2.12 Identificação do sistema. 25
2.13 Indicação de mudança de parâmetros. 26
2.14 Alarme de emergência. 26
2.15 Indicador de transmissão parcial. 26
2.16 Esquema de modulação. 27
2.17 Taxas do código convolucional. 27
2.18 Entrelaçador temporal. 27
2.19 Número de segmentos: possíveis valores e seus significados 28
2.20 Taxa útil de dados em 13 segmentos. 29
2.21 Número de pacotes de 204 *bytes* transmitidos nos 13 segmentos. . 29

3.1 Tabelas de PID's reservados pela norma do MPEG. 34
3.2 Valores possíveis para o adaptation field control. 34
3.3 Valores de PCR extraídos de um TS. 37
3.4 PID's utilizados para transmissão de tabelas do sistema. 38
3.5 Principais Tabelas PSI definidas pelo MPEG. 40
3.6 Principais Tabelas SI utilizadas no sistema ISDB-T_B. 40
3.7 Tipos de receptores e as EIT's associadas. 47
3.8 Tabelas obrigatórias. 47
3.9 Tabelas com aplicações importantes. 47

3.10 Número de pacotes de 204 bytes por quadro de multiplexação. . . 49
3.11 Formação dos 8 bytes de controle. 53
3.12 Formação do Modulation Control configuration Information. . . . 56
3.13 Formação do TMCC Information. 57
3.14 Formação do *Transmission Parameters.* 58

4.1 Principais características do processador Nios II. 69
4.2 Configurações utilizadas nas camadas hierárquicas durante o teste. 96
4.3 Identificação dos equipamentos utilizados no Teste. 106

Lista de Abreviaturas e Siglas

AAC	*Advanced Audio Coding* - Tipo de Codificação de Áudio.
AC	*Ancillary Channel* - Canal Auxiliar.
ABNT	Associação Brasileira de Normas Técnicas.
ASI	*Asynchronous Serial Interface* - Interface Serial Assíncrona.
BOA	*Basic Object Adapter* - Servidor de Páginas de Internet.
BST	*Band Segmented Transmission* - Transmissão de Banda Segmentada.
BTS	*Broadcast Transport Stream* - Fluxo de Transporte de Radiodifusão.
CAT	*Conditional Access Table* - Tabela de Acesso Condicional.
CP	*Continuous Pilot* - Piloto Contínua.
CPU	*Central Processing Unit* - Unidade Central de Processamento.
DBPSK	*Differential Binary Phase Shift Keying* - Chaveamento de Fase Binária Diferencial.
DTS	*Decoding Time Stamps* - Estampas de Tempo de Decodificação.
DVB	*Digital Video Broadcasting* - Radiodifusão de Vídeo Digital.
ES	*Elementary Stream* - Fluxo Elementar.
EIT	*Event Information Table* - Tabela de Infomração de Eventos.
FIFO	*First In First Out* - Tipo de memória onde o primeiro dado a entrar nesta memória também é o primeiro a sair.
FPGA	*Field Programmable Gate Array* - Dispositivo formado por uma matriz de portas lógicas programáveis.
HD	*High-definition* - Alta Definição.

HDTV *High Definition Television* - Televisão de Alta Definição.

IFFT *Inverse Fast Fourier Transform* - Transformada Rápida Inversa de Fourier.

IIP *ISDB-T Information Packet* - Pacote de Informações do ISDB-T.

IP *Internet Protocol* - Protocolo de *Internet*.

ISDB-T *Integrated Services Digital Broadcasting-Terrestrial* - Serviços Integrados de Radiodifusão Digital - Terrestre.

ISDB-T$_B$ *Integrated Services Digital Broadcasting-Terrestrial Brazil* - Serviços Integrados de Radiodifusão Digital - Terrestre Brasil.

LD *Low-definition* - Baixa Definição.

LDTV *Low-definition television* - Televisão de Baixa Definição.

MPEG *Moving Picture Experts Group* - Grupo de Peritos em Imagens Dinâmicas.

NIT *Network Information Table* - Tabela de Informação de Rede.

OFDM *Orthogonal Frequency Division Multiplexing* - Multiplexação por Divisão de Frequências Ortogonais.

PAT *Program Association Table* - Tabela de Associação de Programas.

PCR *Program Clock Reference* - Referência de Relógio para os Programas.

PES *Packetized Elementary Stream* - Fluxo Elementar Empacotado.

PID *Packet Identifier* - Identificador do Pacote.

PSI *Program Specific Information* - Tabelas de Informação dos Programas.

PTS *Presentation Time Stamps* - Estampas de Tempo de Apresentação.

QAM *Quadrature Amplitude Modulation* - Modulação em Amplitude e Quadratura.

QPSK *Quadrature Phase Shift Keying* - Chaveamento de Fase em Quadratura.

RAM *Random Access Memory* - Memória de Acesso Aleatório.

RS *Reed Solomon* - Tipo de Código Corretor de Erro.

RTC *Real Time Clock* - Circuito de Relógio de Tempo Real.

SDT *Service Description Table* - Tabela de Descrição de Serviços.

SDTV *Standard Definition Television* - Televisão de Definição Padrão.

SI *Service Information* - Tabelas de Informação do Sistema.

SP *Scattered Pilots* - Pilotos Espalhadas.

TCP *Transfer Control Protocol* - Protocolo de Controle de Transferência.

TMCC *Transmission and Multiplexing Configuration Control* - Controle e Configuração de Multiplexação e Transmissão.

TOT *Time Offset Table* - Tabela de Mudança de Data e Horário.

TS *Transport Stream* - Fluxo de Transporte.

VHDL *Very High Speed Integrated Circuit Hardware Description Language* - Linguagem de descrição de *Hardware* aplicada a circuitos integrados de alta velocidade.

Lista de Símbolos

A_f	Atraso fim a fim introduzido pelo uso do entrelaçador de *bytes*.
B	Número de ramos do entrelaçador convolucional.
bpp	Número de *bits* transportados por cada portadora.
btx	Quantidade de *bits* transmitidos entre os dois consecutivos pacotes que transportam PCR.
B_{xk}	Número de *bits* que cada camada hierárquica está descarregando em seu seu *buffer*.
D	Atraso inserido em cada símbolo pelo entrelaçador temporal.
I	Comprimento do entrelaçador temporal.
i	Índice do dado no segmento.
ip	Índice do pacote no TS.
k	Representa o número da portadora demodulada.
$mod(a,b)$	Resto da divisão de a por b.
M	Ordem de incremento do número de atrasadores dos ramos do entrelaçador convolucional.
nc	Número total de portadoras por segmento.
n_{pq}	Número de pacotes de 204 *bytes* transmitidos por quadro de multiplexação.
n_{pu}	Número de portadoras úteis.
N_s	Número de segmentos em cada camada hierárquica.
PCR_b	Valor do campo PCR *Base*.
$PCR_{basefinal}$	Valor do PCR *Base* atualizado.
$PCR_{extfinal}$	Valor do PCR *Extension* atualizado.

PCR_e	Valor do campo PCR *Extension*.
pep	Número de pulsos de *clock* entre dois consecutivos pacotes com PCR.
ppq	Número de pacotes transmitidos por quadro OFDM.
R	Taxa útil de transmissão.
Rc	Taxa do fluxo calculada a partir das informações do PCR.
S	Representa o número de *bits* por símbolo da modulação utilizada pela camada hierárquica.
tcd	Taxa de puncionamento.
tep	Intervalo de tempo existente entre consecutivos pacotes com PCR.
Tg	Tempo de duração do intervalo de guarda.
t_q	Tempo de duração do quadro OFDM em segundos.
Tu	Tempo efetivo do símbolo OFDM.
x	Representa a camada hierárquica (A, B ou C).

Resumo

A TV digital promete revolucionar a forma do telespectador assistir televisão. Melhor qualidade de imagem, novas formas de recepção, possibilidade de interatividade e mais opções de programas são algumas das novas possibilidades oferecidas. Para que estas possibilidades se tornassem realidade, foi necessário a utilização de novas teorias, tecnologias e ferramentas. Este trabalho oferece uma referência inicial àqueles que queiram iniciar suas pesquisas na área de TV digital, para compreender seu funcionamento e como estas novas possibilidades são implementadas. Além disso, é proposto um modelo de implementação de um importante equipamento dentro do sistema de transmissão de TV digital que é o multiplexador. Estão mostradas também as tecnologias utilizadas para tal. Estas são referências importantes, uma vez que as normas de TV digital oferecem informações sobre o que o multiplexador deve realizar, mas não há referência de tecnologias que podem ser utilizadas para a sua implementação.

A implantação da TV digital no Brasil e adoção por outros países do mesmo padrão, oferece uma ótima oportunidade de desenvolvimento da indústria e academia brasileira. O trabalho aqui apresentado busca oferecer uma referência e incentivo para que isso aconteça.

Palavras-chave: multiplexador, ISDB-T_B, implementação.

Abstract

The digital TV promises to revolutionize the way how we watch tv. Better images quality, new reception forms, the interactivity possibility, and more programs options are some of the new offered possibilities. New theories and technologies use, and tools were necessary to become those possibilities a reality. This paper offers an initial reference to those who want to start their researches on digital TV, to understand its functioning, and how those new possibilities are implemented. Beyond this, it is proposed an implementation model of an important equipment into the digital transmission system, that is the multiplexer. The used technologies are also shown for that. Those are important references, since the digital TV rules offer information on what the multiplexer must perform, but there are not technologies reference, which can be used for its implementation.

The digital TV introduction in Brazil and other countries adoption in the same standard offer a great industry and Brazilian Academy development opportunity

The present study aims to offer a reference and motivation, so that it happens.

Keywords: Multiplexer, ISDB-T_B, implementation.

Capítulo 1

Introdução

O Brasil entrou em uma nova era da comunicação em massa em Dezembro de 2007, quando se deu o início das transmissões da era da TV digital em alta definição, ou HDTV (*High-Definition Television*), no país. O padrão de TV digital adotado é o ISDB-T_B (*Integrated Services Digital Broadcasting - Terrestrial* Brazil)[1] que é uma evolução do sistema ISDB-T (*Integrated Services Digital Broadcasting - Terrestrial*) [2].

A TV digital promete revolucionar a forma do telespectador assistir à TV. Além do acesso a programação com melhor qualidade de imagem e som, será possível a interação com os programas. A recepção do sinal de TV poderá se dar através de diferentes tipos de receptores: fixos, móveis, portáteis. Além disso, a emissora pode optar pela transmissão de mais de um programa simultaneamente (a chamada multi-programação)[3], aumentando as possibilidades de escolha ao telespectador. Essas novas possibilidades que se apresentam com a TV digital tornam-se possíveis graças a um importante dispositivo que faz parte do sistema de transmissão de TV digital: o multiplexador do sistema ISDB. O multiplexador funciona como uma interface entre o estúdio e o transmissor da emissora de TV. Ele recebe em sua entrada os vários fluxos ou TS's (*Transport Stream*) que a emissora deseja transmitir, os multiplexa com os parâmetros de configuração do transmissor e dados adicionais, como guia de programação ou dados de interatividade. O fluxo entregue na saída do multiplexador, chamado BTS (*Broadcasting Transport Stream*), é repassado ao transmissor. O BTS transporta todas as informações que a emissora deseja transmitir e controla o processo de transmissão.

Pode-se fazer alguns questionamentos a respeito das novas possibilidades oferecidas pela TV digital. Por exemplo, o receptor móvel não possui uma tela com resolução suficiente para exibição de imagens em alta definição (HD -*High-Definition*)com reolução de 1920 colunas $\times$ 1080 linhas [4][5]. Como a emis-

sora pode atender tanto o receptor fixo quanto o móvel, respeitando a resolução disponível em cada sistema? No sistema ISDBT-T_B a solução adotada é a de transmitir um vídeo de alta resolução para o receptor fixo e um outro vídeo com resolução menor para o receptor móvel, o chamado LD (*Low-Definition*), que normalmente possui resolução de 320 × 240. Mas outras perguntas podem surgir. Como o receptor móvel saberá qual o programa enviado à ele? Como o transmissor pode dar maior robustez ao sinal enviado ao celular, já que este estará provavelmente em movimento (o que dificulta a recepção do sinal)? Na multi-programação, como o receptor sabe quantos programas estão no ar? Quantos programas podem ser transmitidos simultaneamente? Como o receptor e o transmissor conseguem separar cada programa um do outro, já que eles serão transmitidos juntos? Estas perguntas são respondidas após estudo da etapa de transmissão e de multiplexação do sistema de TV digital apresentados ao longo deste trabalho.

1.1 Motivação

Um processo de evolução deve ocorrer dentro das emissoras para que estas se adequem à nova era da televisão que se inicia com as transmissões de TV digital em rede aberta no país. Investimentos em novos equipamentos, capacitação de profissionais, adequação de estúdios e melhorias nos cenários das filmagens devem estar envolvidas nesse processo de evolução. Em meio ao processo de implantação da TV digital surge uma oportunidade para a indústria nacional também evoluir. A TV digital deve ser encarada pela indústria brasileira como sendo uma oportunidade de desenvolvimento industrial e tecnológico no país. Essa oportunidade é gerada por uma crescente demanda de equipamentos, treinamento e suporte. A indústria nacional pode favorecer-se à medida que ofereça equipamentos de qualidade e de menor preço que as empresas estrangeiras e a medida que ofereça prontamente aos clientes, o suporte e treinamento necessários. A possibilidade de outros países, principalmente na América Latina, adotarem o mesmo padrão de TV digital que o Brasil, assim como já fizeram a Argentina, Chile e Peru, aumenta ainda mais a oportunidade de crescimento da indústria nacional.

Devido a sua importância dentro do sistema de transmissão, o multiplexador pode ser encarado como uma boa oportunidade de investimento da indústria, uma vez que é um equipamento necessário nas geradoras de TV e nas retransmissoras que desejem realizar inserção de programação local. Neste trabalho, tem-se a apresentação de um modelo proposto para implementação do multiplexador e a apresentação das tecnologias que foram utilizadas. A intenção do trabalho é a de oferecer uma base para pesquisas na área de TV digital e oferecer um ponto

de partida para discussões a respeito de modelos e tecnologias que possam ser utilizadas na implementação do multiplexador.

1.2 Contribuições

Este trabalho apresenta os principais conceitos da etapa de multiplexação relacionando-os com a etapa de transmissão do sistema de TV digital ISDB-T_B. Tais conceitos, apresentados em diversas normas, são mostrados neste trabalho de forma sintetizada. Essa é uma importante contribuição para aqueles que iniciam sua pesquisa na área de TV digital. As informações organizadas formam uma base para que pesquisadores possam rapidamente se interar dessa nova tecnologia e as referências bibliográficas citadas possibilitam o aprofundamento das pesquisas. A maior parte da pesquisa deste trabalho se baseia diretamente nos estudos de normas. As principais normas utilizadas são as normas brasileiras ABNT NBR 15601 [1], ABNT NBR 15603-1 [6], ABNT NBR 15603-2 [7], ABNT NBR 15602-3 [8] e as normas japonesas ARIB (*Association of Radio Industries and Businesses*) STD-B31 [2] e TR-B14 [9], além da norma do MPEG (*Moving Picture Experts Group*) ISO/IEC 13818-1 [10].

Está proposto neste trabalho um modelo para a implementação do multiplexador utilizando tecnologias e ferramentas atuais. As normas de TV digital estabelecem o que o multiplexador deve realizar, mas não oferecem um indicativo a respeito das tecnologias que podem ser empregadas na sua implementação. O modelo proposto oferece uma referência para aqueles que desejem implementar um equipamento multiplexador.

Os principais multiplexadores encontrados no mercado, desenvolvidos em *hardware*, necessitam de um *software* externo para geração de tabelas do sistema. Nesse *software*, o usuário normalmente deve cadastrar manualmente todas as informações necessárias para o correto funcionamento do sistema. Uma característica inovadora do modelo proposto está no fato da geração das principais tabelas como a PAT (*Program Association Table*), PMT (*Program Map Table*) e NIT (*Network Information Table*) ser realizada de maneira automática. O multiplexador se encarrega de analisar os fluxos de entrada e gerar as tabelas necessárias. Uma grande vantagem dessa abordagem é a facilidade de operação do equipamento e o fato de se ter uma grande versatilidade nas alterações do sistema, como reconfigurar algum codificador de vídeo ou alterar alguma informação do sistema de transmissão. Assim que detectada qualquer alteração em alguns desses parâmetros, novas tabelas são geradas levando em consideração sempre as últimas informações.

Uma outra contribuição deste trabalho está na forma de dois artigos publica-

dos tendo como base o conteúdo desta dissertação. Os artigos intitulados "Técnicas para multiplexação de Sinais de TV digital no Padrão ISDB-T_B"e "Solução de Implementação do multiplexador do Sistema ISDB-T_B"foram publicados no Simpósio Brasileiro de Telecomunicações - SBrT - em 2009 e 2008, respectivamente. Estão mostrados maiores detalhes sobre esses artigos no Anexo A.

1.3 Estrutura da dissertação

Este trabalho está estruturado em 5 capítulos. O capítulo 1 traz uma introdução, o capítulo 2 apresenta a camada física do padrão ISDB-T_B, focando nas possibilidades de configurações oferecidas pelo mesmo. O capítulo 3 apresenta detalhes a respeito da formação do fluxo de entrada e saída do multiplexador, que são o TS e o BTS respectivamente. O capítulo 4 mostra detalhes da implementação de um protótipo do multiplexador, evidenciando os desafios da implementação, o modelo adotado, tecnologias e ferramentas utilizadas bem como a validação e testes do protótipo. O capítulo 5 traz as conclusões gerais e as propostas para trabalhos futuros.

Capítulo 2

Camada física do padrão ISDB-T$_B$

Neste capítulo está apresentado o padrão de TV digital adotado no Brasil, o ISDBT-T$_\mathrm{B}$. Abordam-se com maiores detalhes as configurações possíveis do sistema e a análise do impacto da variação desses, principalmente com relação a taxa útil de transmissão dos dados e a transmissão hierárquica, que possibilita a transmissão simultânea para diferentes tipos de receptores (recepção fixa, móvel e portátil). Este capítulo está estruturado da seguinte maneira: a Seção 2.1 apresenta o sistema ISDBT-T$_\mathrm{B}$, a Seção 2.2 detalha o subsistema de transmissão e a Seção 2.3 encerra o capítulo com uma conclusão.

2.1 Sistema Brasileiro de TV digital

O padrão de TV digital adotado no Brasil é uma variante do padrão japonês ISDB-T. As principais diferenças entre esses padrões estão no fato do Brasil utilizar como compressão de vídeo o padrão MPEG-4 (H.264) ao invés do MPEG-2 (H.262) e alterações com relação à máscara de emissão do sinal de radiofrequência.

O sistema ISDB-T é bastante flexível, podendo operar em várias configurações diferentes, pois faz uso de um código corretor de erros puncionado, possibilita a utilização de entrelaçadores (*interleaving*) no domínio do tempo e da frequência e utiliza a técnica de modulação OFDM (*Orthogonal Frequency Division Multiplexing*). O sistema faz uso da segmentação da banda de transmissão (BST OFDM - *Band Segmented Transmission* OFDM) e possibilita a transmissão hierárquica de até três camadas chamadas de camadas A, B e C. Deve-se escolher, para cada uma dessas camadas, os parâmetros adequados à taxa de transmissão e à robustez desejada.

Tem-se, na Figura 2.1, uma visão resumida do sub-sistema de modulação do ISDB-T$_\mathrm{B}$ e os seus pontos de configuração. Na figura estão evidenciados o re-multiplexador de TS e a transmissão hierárquica do sub-sistema.

Figura 2.1: *Sub-sistema de modulação de TV digital.*

O multiplexador é a etapa responsável por entregar um único fluxo à entrada do transmissor e também por configurar os parâmetros de transmissão do mesmo. Na Figura 2.2 pode-se visualizar o multiplexador recebendo vários TS's em sua entrada, mais os dados e configurações. O BTS, como é chamado o fluxo de saída do multiplexador, é encaminhado ao transmissor que será configurado segundo os parâmetros definidos por esse mesmo fluxo.

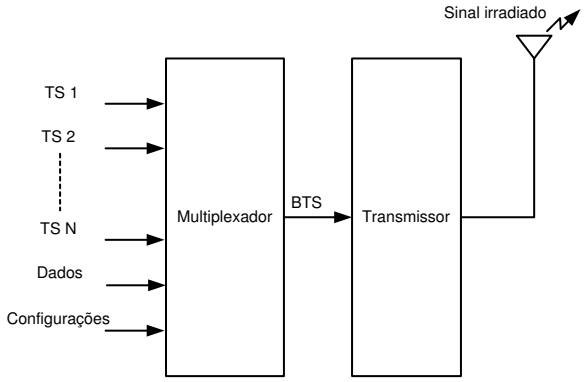


Figura 2.2: *Sub-sistema de transmissão e o multiplexador.*

2.2 Sub-sistema de modulação

A Figura 2.3 mostra uma visão completa do sub-sistema de transmissão do padrão ISDB-T_B. Conforme já comentado, esse é idêntico ao do padrão ISDB-T.

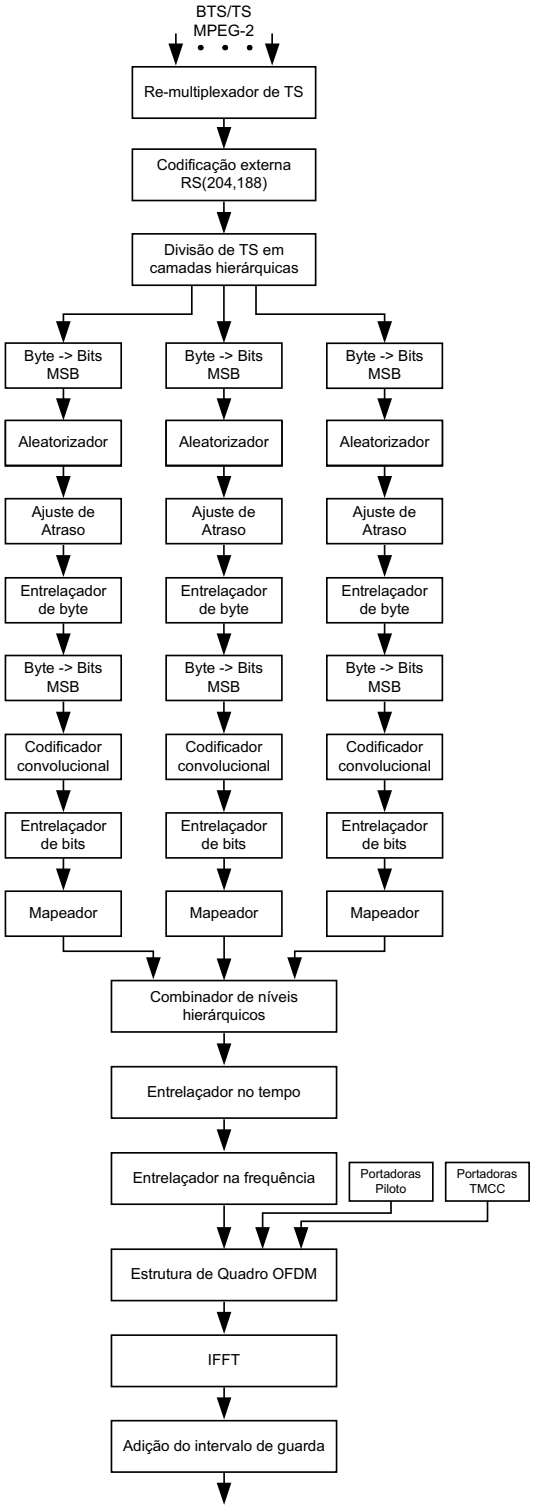


Figura 2.3: *Diagrama em blocos do sub-sistema de modulação*

A seguir, é dada uma descrição sobre a função de cada etapa do sub-sistema de modulação.

2.2.1 Re-multiplexador

O re-multiplexador é primeira etapa do sub-sistema de modulação, sendo responsável por receber o fluxo de dados a ser transmitido e decidir corretamente em qual camada um dado pacote deve ser transmitido. Essa decisão ocorre entre o descarte ou o encaminhamento à uma das três camadas hierárquicas do transmissor. O fluxo de entrada pode ser um TS MPEG-2 (fluxo geralmente formado por pacotes de 188 *bytes*) ou um BTS (fluxo formado por pacotes de 204 *bytes*). Cada pacote, antes de ser encaminhado ao seu destino, passa por um codificador *Reed-Solomon* RS(204,188), que insere 16 *bytes* de paridade a cada 188 *bytes*. A Figura 2.4 ilustra o funcionamento do re-multiplexador descrito acima, em conjunto com o codificador Reed-Solomon, classificado como codificador externo.

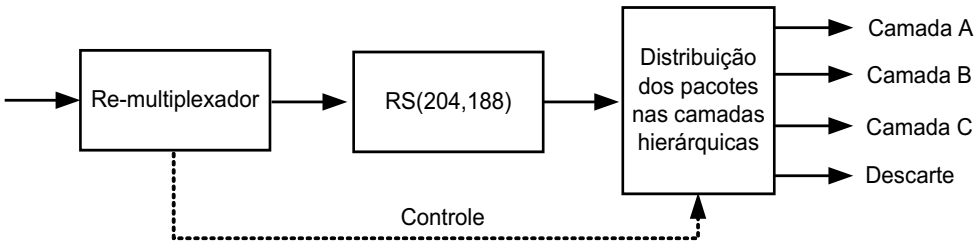


Figura 2.4: *O re-multiplexador e o codificador RS.*

Transmissão de um TS

Caso o fluxo a ser transmitido tenha sido entregue por um codificador de vídeo diretamente conectado à entrada do transmissor, tem-se na entrada do re-multiplexador um TS. Nesse caso, o transmissor normalmente opera em uma configuração padrão pré-definida pelo fabricante.

O re-multiplexador deverá organizar corretamente o fluxo a ser entregue às próximas etapas de forma a atender as regras de formação do quadro de multiplexação. A formação do quadro de multiplexação será mostrada na Sessão 3.3.

A cada pacote do TS, formado por 188 *bytes*, são inseridos 16 *bytes* de paridade pelo codificador *Reed-Solomon* e em seguida o pacote é encaminhado a uma das camada hierárquicas de transmissão.

Transmissão de um BTS

Caso o fluxo a ser transmitido tenha sido entregue por um multiplexador, tem-se um BTS na entrada do re-multiplexador. As configurações do transmissor serão recebidas através de um pacote especial chamado de IIP (ISDB-T *Information Packet*).

Cada pacote do BTS, formado por 204 *bytes*, possui uma marcação especial que indica para qual camada o pacote deve ser encaminhado. Essa marcação é enviada, juntamente com outras informações, nos 16 *bytes* finais de cada pacote [1],[2]. Estes 16 *bytes* são substituídos pelos *bytes* de paridade do codificador *Reed-Solomon* antes de serem repassados às etapas seguintes.

2.2.2 Codificador Reed-Solomon

O codificador *Reed-Solomon* RS(204, 188) adiciona 16 *bytes* de paridade a cada pacote de 188 *bytes* a ser transmitido [1],[2]. Os *bytes* de paridade serão responsáveis pela proteção do bloco codificado, fazendo com que seja possível a detecção e correção de erros durante a decodificação dos dados [11],[12]. Na Figura 2.5 é possível visualizar a inserção dos *bytes* de paridade do RS em um pacote do TS.

Figura 2.5: *Pacote MPEG acrescido de 16 bytes do RS.*

2.2.3 Aleatorizador

O aleatorizador realiza uma operação ou-exclusivo *bit* a *bit* entre os dados que lhe são entregues e uma sequência pseudo-aleatória. Essa operação garante um espalhamento de energia do sinal no domínio da frequência, uma vez que torna os dados em sua saída com características aleatórias [11],[12].

Quanto mais concentrada em pequenas faixas de frequência for a potência do sinal, mais susceptível será o mesmo às interferências de faixa estreita (*Jammimg*) ou a desvanecimento seletivo causado por multipercursos. O processo de aleatorização reduz a influência desses fenômenos no sinal recebido.

Na Figura 2.6 visualiza-se o circuito gerador da sequência pseudo-aleatória utilizado no aleatorizador e seu valor inicial no princípio de cada quadro de multiplexação. Todos os dados a serem transmitidos, com exceção do *byte* de sincronismo de cada pacote MPEG, devem sofrer a operação ou-exclusivo com a sequência pseudo-aleatória antes de serem repassados às etapas seguintes.

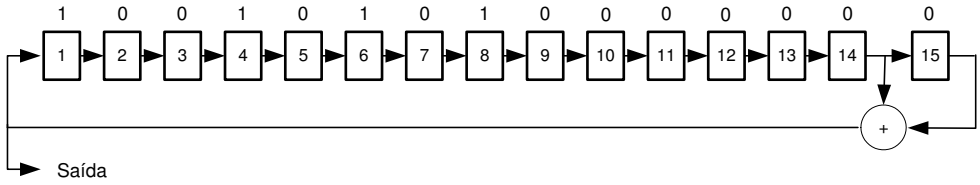


Figura 2.6: *Circuito gerador da sequência pseudo-aleatória.*

2.2.4 Ajuste de atraso

Essa etapa tem o objetivo de fazer com que o atraso inserido pelo entrelaçador de *bytes* seja idêntico entre as diferentes camadas do sistema. De acordo com a configuração de cada camada, um determinado atraso deverá ser inserido, como mostra a Tabela 2.1, onde N_s é o número de segmentos em cada camada hierárquica [1],[2].

Tabela 2.1: *Ajuste de atraso.*

Mapeamento	Taxa do Código Convolucional	Ajuste de atraso em número de pacotes de 204 *bytes*		
		Modo 1	Modo 2	Modo 3
QPSK/DQPSK	1/2	12 × N_s-11	24 × N_s-11	48 × N_s-11
	2/3	16 × N_s-11	32 × N_s-11	64 × N_s-11
	3/4	18 × N_s-11	36 × N-11	72 × N_s-11
	5/6	20 × N_s-11	40 × N-11	80 × N_s-11
	7/8	21 × N_s-11	42 × N_s-11	84 × N_s-11
16QAM	1/2	24 × N_s-11	48 × N_s-11	96 × N_s-11
	2/3	32 × N_s-11	64 × N_s-11	128 × N_s-11
	3/4	36 × N_s-11	72 × N_s-11	144 × N_s-11
	5/6	40 × N_s-11	80 × N_s-11	160 × N_s-11
	7/8	42 × N_s-11	84 × N_s-11	168 × N_s-11
64QAM	1/2	36 × N_s-11	72 × N_s-11	144 × N_s-11
	2/3	48 × N_s-11	96 × N_s-11	192 × N_s-11
	3/4	54 × N_s-11	108 × N_s-11	216 × N_s-11
	5/6	60 × N_s-11	120 × N_s-11	240 × N_s-11
	7/8	68 × N_s-11	126 × N_s-11	252 × N_s-11

2.2.5 Entrelaçador de *bytes*

O entrelaçador atua em conjunto com códigos corretores de erro e ajudam a melhorar o desempenho do sistema de transmissão, principalmente em canais onde ocorrem erros em rajadas, ou seja, com ocorrência de desvanecimentos e

ruído impulsivo. A função do entrelaçador é embaralhar os símbolos entregues pelo codificador, fazendo com que a correlação entre os símbolos de uma mesma palavra código diminua [13].

O entrelaçador usado no sistema ISDB-T é do tipo convolucional e é composto por B ramos. Cada ramo possui um determinado número de atrasadores e o número de atrasadores por ramo segue uma determinada ordem de incremento M. O primeiro ramo tem 0 atrasadores, o segundo tem M atrasadores, o terceiro ramo $2\times M$ atrasadores, e assim por diante, até o último ramo onde se tem (B-1)$\times M$ atrasadores. O entrelaçador do sistema ISDB-$\mathrm{T_B}$, mostrado na Figura 2.7, possui M=17 e B=12 [1],[2].

Figura 2.7: *Estrutura do entrelaçador de bytes.*

O atraso inserido em *bytes* devido a utilização do conjunto entrelaçador / desentrelaçador, também chamado de atraso fim a fim A_f, é dado por

$$A_f = N \times (B - 1), \tag{2.1}$$

onde N é determinado por:

$$N = B \times M. \tag{2.2}$$

No caso do sistema ISDB-$\mathrm{T_B}$, tem-se N igual a 204, resultando em um atraso total de 2.244 *bytes*. O atraso fim a fim representa o atraso inserido até que desentrelaçador entregue os dados desentrelaçados em sua saída. No total, o atraso será de 11 pacotes de 204 *bytes* [1],[2].

2.2.6 Codificador convolucional

Essa etapa engloba um código convolucional de taxa fixa, chamado de código mãe, e também um puncionador que permite variar a taxa de codificação do sistema. O código mãe utilizado no sub-sistema de modulação possui taxa de codificação igual a 1/2. O polinômio gerador do codificador é dado por $G_1 = 171_o = 1111001_b$

e $G_2 = 133_o = 1011011_b$ [1],[2]. A Figura 2.8 ilustra o diagrama em blocos do codificador de taxa 1/2, cuja profundidade de código é igual a 7. Nesse codificador, os *bits* de dados a serem codificados alimentam uma sequência de registradores. O codificador possui duas saídas, X e Y, definidas pelas operações ou-exclusivo realizadas entre a saída de alguns registradores, em função de G_1 e G_2.

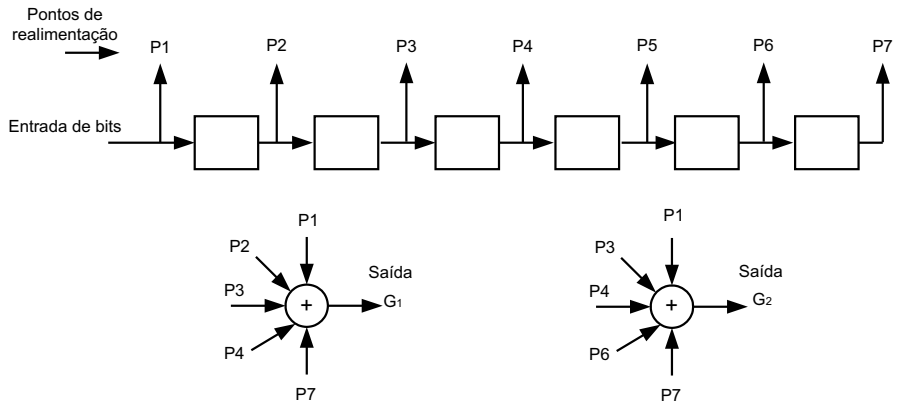


Figura 2.8: *Diagrama em blocos do codificador convolucional com taxa 1/2.*

No sistema ISDB-T_B é possível configurar a taxa de codificação com os valores de 1/2, 2/3, 3/4, 5/6 e 7/8. Aumentando-se a taxa do código, diminui-se o número de *bits* enviados, reduzindo a proteção contra erros. As diferentes taxas de códigos são obtidas a partir do uso de uma técnica denominada de puncionamento, que consiste em excluir alguns *bits* da saída do codificador. As regras de puncionamento para as diferentes taxas citadas estão mostradas na Tabela 2.2, onde pode-se visualizar quais informações recebidas do código (X e Y) são repassadas às etapas seguintes. Na recepção utiliza-se um decodificador configurado para taxa 1/2 e há inserção de *bits* aleatoriamente, na entrada do decodificador, nas posições dos dados que não foram transmitidos devido ao processo de puncionamento.

Tabela 2.2: *Puncionamento.*

Taxa de código	Padrão de Puncionamento	Sequência de transmissão
1/2	X:1 / Y:1	X1,Y1
2/3	X:1,0 / Y:1,1	X1,Y1,Y2
3/4	X:1,0,1 / Y:1,1,0	X1,Y1,Y2,X3
5/6	X:1,0,1,0,1 / Y:1,1,0,1,0	X1,Y1,Y2,X3,Y4,X5
7/8	X:1,0,0,0,1,0,1 / Y:1,1,1,1,0,1 0	X1,Y1,Y2,Y3,Y4,X5,Y6,X7

2.2.7 Mapeamento das portadoras

Os dados que irão ser transmitidos em cada portadora do sinal OFDM são mapeados previamente, de acordo com as configurações de transmissão, nas constelações DQPSK, QPSK, 16QAM ou 64QAM. Porém, antes de serem mapeados, os dados passam por um ajuste de atraso e por um entrelaçador de *bits*, como ilustra a Figura 2.9 [1],[2].

Figura 2.9: *Processo de mapeamento.*

Ajuste de atraso de *bits*

O ajuste de atraso de *bits* tem como função fazer com que o atraso devido a utilização do entrelaçador de *bit* seja idêntico entre as camadas hierárquicas.

O entrelaçador de *bits* irá inserir um atraso de 120 símbolos mapeados, considerando o atraso inserido na transmissão e na recepção. Porém, de acordo com o esquema de modulação selecionado para a camada, são necessários tempos diferentes para se obter o atraso especificado de 120 símbolos. O ajuste de atraso de *bits* fixa o atraso das diferentes camadas em 2 símbolos OFDM, independentemente da configuração de cada camada [1],[2]. Na Tabela 2.3 tem-se o valor de atraso em *bits* a ser inserido de acordo com o modo de transmissão, com o mapeamento e com o número de segmentos (N_s) de cada camada. O modo de transmissão está relacionado com o número de portadoras que formam cada símbolo OFDM transmitido. Na Seção 2.2.12 tem-se maiores detalhes sobre os modos de transmissão.

Tabela 2.3: *Ajuste de atraso em bits.*

Mapeamento	Modo 1	Modo 2	Modo 3
QPSK	(384×N_s)-240	(768×N_s)-240	(1536×N_s)-240
16QAM	(768×N_s)-480	(1536×N_s)-480	(3072×N_s)-480
64QAM	(1152×N_s)-720	(2304×N_s)-720	(4608×N_s)-720

Estrutura do entrelaçador de *bits*

O seletor 1 da Figura 2.9 encaminha os *bits* ao entrelaçador agrupados de acordo com o mapeamento selecionado. Por exemplo, repassa de dois em dois *bits* caso o mapeamento selecionado seja o DQPSK ou QPSK, de quatro *bits* caso seja o mapeamento 16QAM e de seis *bits* no caso do 64QAM.

Cada mapeamento possui uma estrutura de entrelaçador correspondente, como pode-se conferir na Figura 2.10. O *bit* mais significativo é encaminhado ao ramo sem atraso.

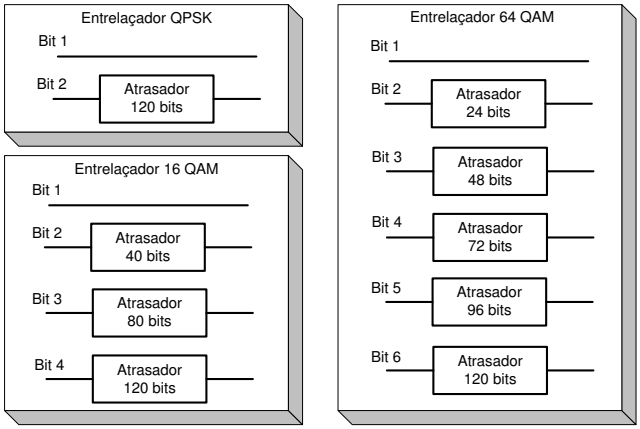


Figura 2.10: *Estrutura dos entrelaçadores de bits*

Mapeamento de *bits*

Os *bits* da saída do entrelaçador são mapeados segundo a modulação selecionada. No caso do mapeamento usando a modulação QPSK, cada dois *bits* são convertidos em pontos da constelação apresentada na Figura 2.11. No mapeamento 16QAM quatro *bits* são convertidos em um ponto da constelação apresentada na Figura 2.12. Para o 64QAM, seis *bits* são convertidos em um dos pontos da constelação da Figura 2.13.

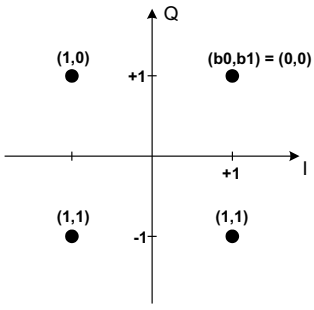


Figura 2.11: *Constelação QPSK.*

Figura 2.12: *Constelação 16QAM.*

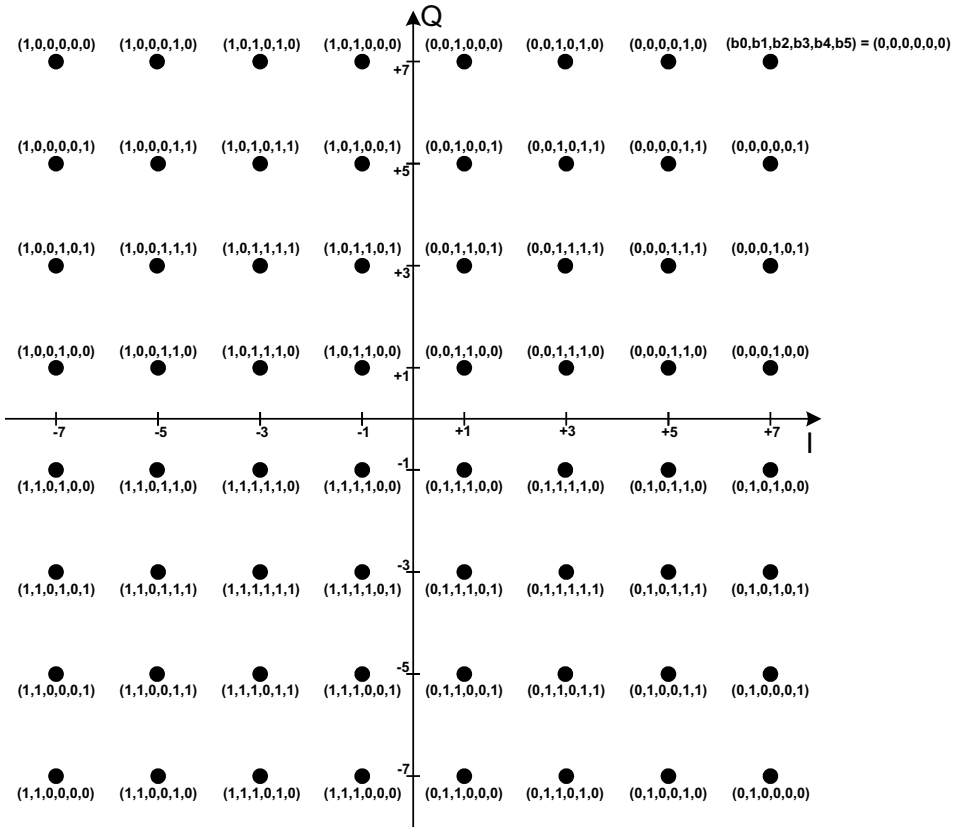


Figura 2.13: *Constelação 64QAM.*

2.2.8 Segmentos de dados

O sinal a ser transmitido é formado por uma sucessão de símbolos OFDM e cada um desses símbolos é formado por um conjunto de portadoras de dados e

portadoras do sistema.

Cada símbolo OFDM é formado por subconjuntos de portadoras, chamados de segmentos. Independente do modo de transmissão, o símbolo OFDM é formado por 13 segmentos. A Tabela 2.4 mostra o número de portadoras de dados úteis por segmento para cada modo de operação.

Tabela 2.4: *Número de portadoras de dados úteis de acordo com o modo.*

Modo	k portadoras/segmento	Total de portadoras
1	96	1248
2	192	2496
3	384	4992

A Figura 2.14 ilustra a formação dos segmentos em um símbolo OFDM.

Figura 2.14: *O símbolo OFDM e os segmentos de dados*

Cada uma das portadoras de dados transporta um símbolo resultante do processo de mapeamento de *bits*.

2.2.9 Combinador de níveis hierárquicos

No combinador, os dados são organizados para formarem os segmentos de dados. Os símbolos complexos resultantes do processo de mapeamento das diferentes camadas de transmissão são combinados de forma a entregar os dados na ordem necessária para se gerar um símbolo OFDM.

Cada uma das camadas do transmissor é responsável por um número configurável de segmentos. Essa configuração é válida desde que o número de segmentos resultante da soma dos segmentos das camadas A, B e C seja igual a 13. A camada A é responsável pelos primeiros segmentos do símbolo OFDM, seguidos pelos segmentos da camada B e, por fim, os segmentos da camada C. Na Figura 2.15 visualiza-se a organização dos dados pelo combinador, considerando que o sistema de transmissão esteja configurado com 1 segmento na camada A,

7 segmentos na a camada B e 5 segmentos na a camada C. Os segmentos foram representados pela letra S e o símbolo mapeado que irá ser transportado por uma portadora do segmento foi representado pela letra P.

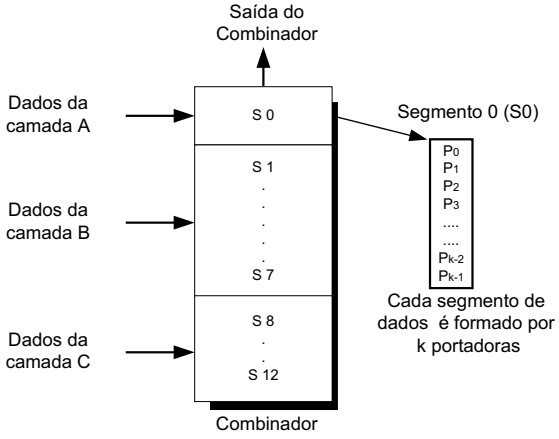


Figura 2.15: *Exemplo de organização dos dados no combinador.*

2.2.10 Entrelaçador Temporal

O entrelaçador temporal embaralha os símbolos mapeados através da inserção de diferentes atrasos em cada símbolo. A estrutura do entrelaçador temporal está ilustrada na Figura 2.16.

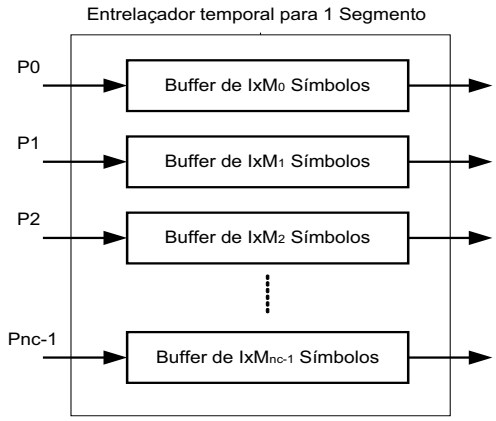


Figura 2.16: *Estrutura do entrelaçador temporal.*

Na figura, cada símbolo mapeado foi representado pela letra P. Cada um desses símbolos, de acordo com sua posição dentro do segmento, sofre um determinado atraso D que é definido por

$$D = I \times Mi, \tag{2.3}$$

onde I representa o comprimento do entrelaçador temporal e Mi é dado por

$$Mi = mod((i \times 5), nc). \tag{2.4}$$

Sendo que i representa o índice do dado no segmento, nc é o número total de portadoras por segmento e $mod(a,b)$ representa o resto da divisão de a por b. O valor de I varia de acordo com a configuração do sistema e pode assumir diferentes valores de acordo com o modo de transmissão. Também é importante ressaltar que nc assume um valor pré-determinado para cada modo de transmissão, conforme mostra a Tabela 2.5.

Tabela 2.5: *Possíveis valores de I e nc de acordo com o Modo.*

	Modo 1	Modo 2	Modo 2
I	0, 4, 8 e 16	0, 2, 4 e 8	0, 1, 2 e 4
nc	96	192	384

Cada camada de transmissão possui seu entrelaçador temporal e cada um deles pode ser configurado com um valor de comprimento diferente. A diferença de tempo nos dados das diferentes camadas devido a utilização do entrelaçador temporal deve ser compensada na transmissão. O atraso é compensado de forma que o atraso total na transmissão e recepção seja múltiplo do tempo de duração do quadro OFDM, que é formado por 204 símbolos OFDM. Na Tabela 2.6 são mostradas as possíveis configurações para o entrelaçador temporal e os respectivos atrasos que devem ser inseridos antes do entrelaçador. O atraso inserido é dado em número de símbolos mapeados.

2.2.11 Entrelaçador no domínio da frequência

Nesta etapa, os dados são processados de acordo com o tipo de modulação selecionada para cada segmento e os dados referentes a recepção parcial, quando existir, são processados separadamente. A recepção parcial refere-se aos dados do primeiro segmento, quando este é configurado para transmissão para receptores móveis. Os tipos de modulação podem ser diferencial, para o caso do segmento transportar símbolos mapeados utilizando-se DQPSK, ou pode ser do tipo coerente, para o caso do segmento transportar símbolos mapeados utilizando-se QPSK, 16QAM ou 64QAM.

O diagrama em blocos do entrelaçador na frequência pode ser visualizado na Figura 2.17. Numa primeira etapa, os dados pertencentes a um mesmo tipo de modulação, porém pertencentes a segmentos diferentes são embaralhados entre si. Esses dados embaralhados são agrupados de forma a manter a organização

Tabela 2.6: *Comprimentos do entrelaçador temporal e atrasos necessários.*

	Comprimento do entrelaçador (I)	Número de símbolos do ajuste atraso	Número de quadros de atraso na transmissão e na recepção
Modo 1	0	0	0
	4	28	2
	8	56	4
	16	112	8
Modo 2	0	0	0
	2	14	1
	4	28	2
	8	56	4
Modo 3	0	0	0
	1	109	1
	2	14	1
	4	28	2

de segmentos (os símbolos embaralhados são agrupados de acordo com o número de portadoras por segmento) e seguem para a etapa seguinte. Nessa, os dados pertencentes a um mesmo segmento são embaralhados entre si. Note que os segmentos referidos na segunda etapa são os segmentos gerados pela primeira etapa.

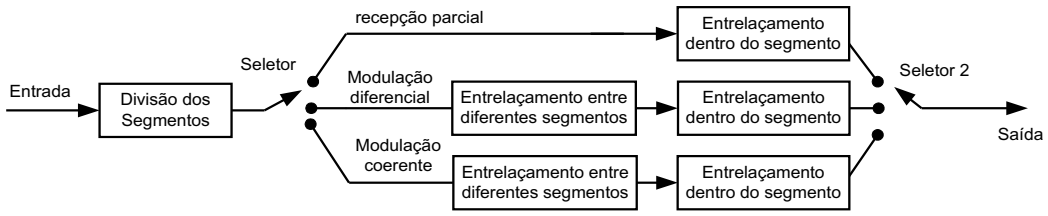


Figura 2.17: *Entrelaçador no domínio da frequência.*

2.2.12 Sinal OFDM

Como citado na sessão 2.1, o sistema de transmissão utiliza segmentação em banda e a técnica OFDM. A técnica OFDM faz uso de múltiplas portadoras ortogonais enviadas simultaneamente no canal de transmissão. No sistema ISDB-T_B é possível configurar a quantidade de portadoras que irão compor os símbolos OFDM transmitidos [1],[2], tal como descrito a seguir:

- Modo 1: 2048 portadoras com espaçamento de 3,968 kHz entre si; do total,

1405 portadoras transportam dados úteis e informações do sistema.

- Modo 2: 4096 portadoras com espaçamento de 1,9841 kHz entre si; dessas, 2809 portadoras transportam dados úteis e informações do sistema.
- Modo 3: 8192 portadoras com espaçamento de 0,9920 kHz entre si; dessas, 5617 portadoras transportam dados úteis e informações do sistema.

As portadoras que não transportam dados úteis e informações do sistema transmitem informação nula e são necessárias para completar o total de portadoras da IFFT.

Também há a possibilidade de se configurar o intervalo de guarda utilizado no sistema, que é uma cópia da parte final de um símbolo OFDM transmitido no início daquele mesmo símbolo, conforme está mostrado na Figura 2.18 [14]. O tempo efetivo de símbolo (Tu) é obtido na saída do bloco da IFFT que é a responsável pela geração do sinal OFDM. Parte deste símbolo é copiada para a geração do intervalo de guarda (Tg) resultando no símbolo OFDM a ser transmitido que terá duração de $Tu + Tg$. O quadro OFDM por sua vez, formado por 204 símbolos OFDM, terá duração 204 vezes maior que a duração de um símbolo OFDM, conforme mostra a Tabela 2.7. Pode-se configurar a duração do intervalo de guarda selecionando a fração do tempo de símbolo OFDM efetivo que será copiado para formar o intervalo de guarda, sendo que as opções disponíveis no sistema são 1/4, 1/8, 1/16 e 1/32. O intervalo de guarda aumenta a robustez do sinal frente a múltiplos percursos introduzidos pelo canal [15].

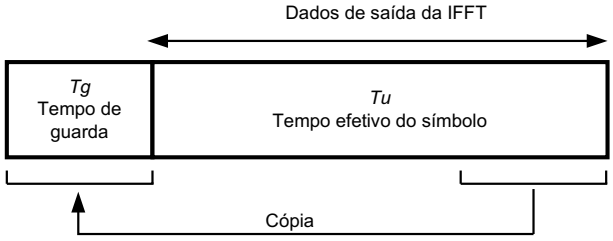


Figura 2.18: *O intervalo de guarda.*

A banda de transmissão, que é de 6 MHz no caso do Brasil e do Japão, é dividida em 14 segmentos iguais. Destes 14 segmentos, 13 são utilizados para transmissão e 1 como banda de guarda [14]. Portanto, cada segmento possui largura de aproximadamente 430 kHz e a banda de transmissão utilizada pelos 13 segmentos é de aproximadamente 5,6 MHz. Na Figura 2.19 é possível visualizar a divisão do espectro nos 13 segmentos e a formação de cada segmento. Cada segmento, de acordo com o modo de transmissão, será formado por um determinado número de portadoras. Para os modos 1, 2 e 3 tem-se 108, 216 e

Tabela 2.7: *Duração do símbolo e do quadro OFDM.*

Modo	1			
Tu - Tempo de efetivo de símbolo (μs)	252			
Intervalo de guarda	1/4	1/8	1/16	1/32
Tg - Duração do Intervalo de Guarda (μs)	63	31,5	15,75	7,875
Duração do símbolo OFDM (μs)	315	283,5	267,75	259,875
Duração do quadro OFDM (ms)	64,26	57,834	54,621	53,0145
Modo	2			
Tu - Tempo de efetivo de símbolo (μs)	504			
Intervalo de guarda	1/4	1/8	1/16	1/32
Tg - Duração do Intervalo de Guarda (μs)	126	63	31,5	15,75
Duração do símbolo OFDM (μs)	630	567	535,5	519,75
Duração do quadro OFDM (ms)	128,52	115,668	109,242	106,029
Modo	3			
Tu - Tempo de efetivo de símbolo (ms)	1,008			
Intervalo de guarda	1/4	1/8	1/16	1/32
Tg - Duração do Intervalo de Guarda (μs)	252	126	63	31,5
Duração do símbolo OFDM (ms)	1,26	1,134	1,071	1,0395
Duração do quadro OFDM (ms)	257,04	231,336	218,484	212,058

432 portadoras por segmento, respectivamente, incluindo as portadoras de dados e outras portadoras especiais. Uma dessas portadoras especiais é utilizada como referência no processo de demodulação está identificada na Figura 2.19 como portadora externa.

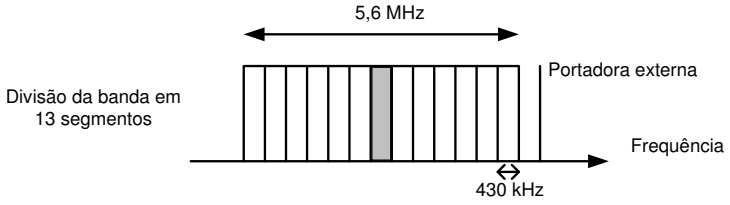


Figura 2.19: *Segmentação da banda e formação de cada segmento.*

Um exemplo de configuração utilizada no início das transmissões no Brasil foi o modo 3, intervalo de guarda 1/16 e com parâmetros das camadas hierárquicas semelhantes aos mostrados na Tabela 2.8.

A Figura 2.20 mostra o agrupamento dos segmentos, considerando-se o sinal em RF, para a camada A configurada para 1 segmento e a camada B para 12 segmentos. Na Figura 2.21 visualiza-se o agrupamento dos segmentos para a camada A com 1 segmento, camada B com 8 segmentos e camada C com 4 segmentos. Para a transmissão em RF, os segmentos são organizados de forma

Tabela 2.8: *Exemplo de configuração utilizada nas transmissões iniciais no Brasil.*

	Número de segmentos	Taxa de codificação	Modulação	Taxa de vídeo transmitido
Camada A	1	2/3	QPSK	LD com taxa por volta de 400 kbps
Camada B	12	3/4 ou 5/6	64QAM	HD com taxa por volta de 15 Mbps
Camada C	0	–	–	–

que o segmento destinado a transmissão parcial ocupe o segmento central do espectro.



Figura 2.20: *Exemplo de transmissão hierárquica: duas camadas ativas.*



Figura 2.21: *Exemplo de transmissão hierárquica: três camadas ativas.*

Formação do símbolo e do quadro OFDM

O símbolo OFDM é formado por portadoras que transportam os dados a serem transmitidos, mais portadoras especiais que transportam informações adicionais do sistema. As portadoras especiais são classificadas em SP's (Pilotos espalhadas - *Scattered Pilot*), CP's (Pilotos contínuas - *Continual Pilot*), AC's (canal auxiliar - *Auxiliary Channel*) e TMCC's (portadoras de parâmetros de transmissão, multiplexação, controle e configuração - *Transmission and Multiplexing Configuration Control*).

As portadoras TMCC's possuem um papel de destaque no sistema, uma vez que é através delas que são transmitidos os parâmetros de transmissão do sistema

como o número de segmentos por camada, o esquema de modulação, a taxa de códigos e a profundidade do entrelaçador temporal. Além disso, as portadoras TMCC são utilizadas na sincronização do quadro OFDM[3][14].

Na Tabela 2.9 é possível observar a relação de portadoras úteis e portadoras especiais, bem como a distribuição da quantidade de cada um dos tipos de portadoras especiais dentro de um segmento OFDM, considerando que o sistema esteja utilizando um esquema de modulação coerente, ou seja QPSK, 16QAM ou 64QAM.

Tabela 2.9: *Números e tipos de portadoras do padrão ISDB-T_B.*

Portadoras por Segmento			
Modo	1	2	3
Total de portadoras	108	216	432
Dados	96	192	384
SP	9	18	36
CP	0	0	0
TMCC	1	2	4
AC	2	4	8
Portadoras por Símbolo OFDM			
Modo	1	2	3
Total de portadoras	1405	2809	5617
Dados	1248	2496	4992
SP	117	234	468
CP	1	1	1
TMCC	13	26	52
AC	26	52	104

As portadoras AC e TMCC são transmitidas utilizando-se a modulação DBPSK (*Differential Binary Phase Shift Keying*), enquanto que na transmissão das CP e SP utiliza-se a modulação BPSK.

As portadoras TMCC's

Essas portadoras possuem um papel importante dentro do sistema pois transportam todos os parâmetros de configurações de transmissão, como mencionado na Seção 2.2.12. Os parâmetros de configuração devem ser gerados pelo multiplexador, que é o responsável pela configuração do transmissor, e devem estar de acordo com as regras de formação e de sinalização das portadoras TMCC. De acordo com a configuração do modo de transmissão, tem-se um determinado número de portadoras TMCC transmitidas por segmento, conforme mostrado na Tabela 2.9.

As portadoras TMCC pertencentes a um mesmo símbolo OFDM transportam a mesma informação. Portando, independente do modo e, consequentemente, do número de TMCC transmitidas por segmento OFDM, 204 símbolos ou *bits* serão transmitidos pelas portadoras TMCC por quadro OFDM, conforme mostrado na Figura 2.22.

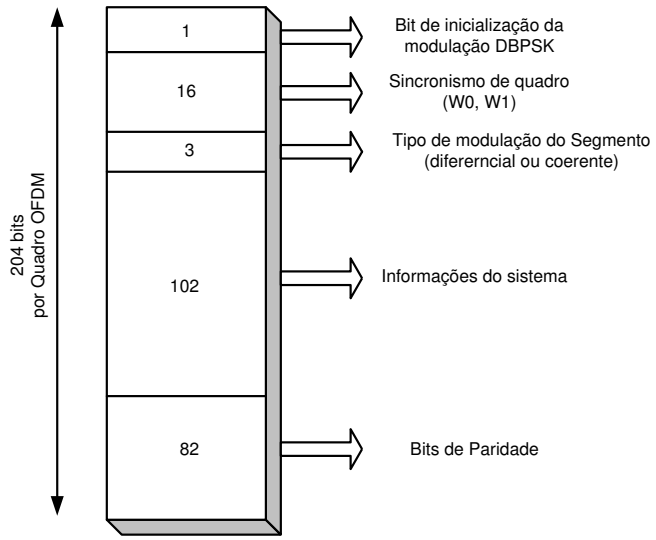


Figura 2.22: *Formação, em bits, do quadro da TMCC.*

O primeiro *bit* é uma referência de amplitude e da fase para a demodulação diferencial, que deverá ser utilizada para demodulação dos dados da TMCC. O sincronismo de quadro entre o transmissor e o receptor é realizado através de uma palavra de 16 *bits* que tem seu valor alternado em cada quadro OFDM. Ora assume o valor w_0, ora w_1, sendo que: w_0= 0011010111101110 e
w_1= 1100101000010001. O tipo da modulação utilizado por segmento é indicado através de 3 *bits*, sendo que assumem o valor 111 para modulação diferencial e 000 para modulação coerente. Os 102 *bits* de informação do sistema transportam as configurações e serão utilizados na recepção para o receptor poder se configurar da mesma maneira que o transmissor e, assim, decodificar corretamente o sinal recebido. Esses *bits*, como mostrado na Tabela 2.10, transportam as configurações atuais do sistema (*current information*), as próximas (*next information*) além de um contador decrescente para sincronizar alteração de parâmetros.

Nas Tabelas 2.11, 2.12 e 2.13 têm-se os parâmetros referentes a cada camada, os valores possíveis e seus respectivos significados para o campo identificação do sistema e os possíveis valores para a indicação de mudança de parâmetros. A mudança de configuração do sistema não acontece de forma instantânea e a transição de parâmetros no transmissor e no receptor deve acontecer no mesmo instante. A transição de uma configuração chamada de atual para a chamada

Tabela 2.10: *TMCC - Os 102 bits de informações do sistema.*

bit da TMCC	Descrição
20 ao 21	Identificação do sistema
22 ao 25	Indicação de mudança de parâmetros
26	Iniciar alarme de emergência
Configurações atuais ou correntes	
27	Indicação de recepção parcial
28 ao 40	Configurações da camada A
41 ao 53	Configurações da camada B
54 ao 66	Configurações da camada C
Próximas configurações	
67	Indicação de recepção parcial
68 ao 80	Configurações da camada A
81 ao 93	Configurações da camada B
94 ao 106	Configurações da camada C
Reservados	
107 ao 109	Reservados (para rádio digital)
110 ao 121	Reservados

próxima se dá somente após 15 quadros OFDM contados a partir do pedido de alteração de parâmetros. Um contador decrescente, que é transportado no campo indicação de mudança de parâmetros, contabiliza quantos quadros OFDM faltam para a efetiva troca de parâmetros.

Tabela 2.11: *Descrição da configuração de cada camada.*

Configurações da camada	Número de *bits*
Esquema de modulação	3
Taxa do código convolucional	3
Tempo do entrelaçador temporal	3
Número de segmentos	4

Tabela 2.12: *Identificação do sistema.*

Valor (*bits* 20 e 21)	Significado
00	Sistema baseado na norma STD-B31 / ABNT NBR 15601 que especifica o sistema ISDB-T /ISDB-T_B
01	Sistema de rádio digital do ISDB
10, 11	Reservado

Tabela 2.13: *Indicação de mudança de parâmetros.*

Valor (*bits* 22 ao 25)	Significado
1111	Valor normal
1110	Faltam 15 quadros para troca de parâmetros
1101	Faltam 14 quadros para troca de parâmetros
...	...
0000	Falta 1 quadro para troca de parâmetros

No sistema ISDB foi previsto um sistema de envio de alarme para todos os receptores de TV em caso de uma emergência, como por exemplo um terremoto, que é comum no Japão. O envio desse alarme faz uso do envio de sinalização em uma portadora, que é o indicador de alarme de emergência. A Tabela 2.14 mostra os valores e significados para este campo.

Tabela 2.14: *Alarme de emergência.*

Valor (*Bit* 26)	Significado
0	Alarme de emergência desativado
1	Alarme de emergência ativado

O indicador de transmissão para receptores portáteis pode assumir os valores indicados na Tabela 2.15.

Tabela 2.15: *Indicador de transmissão parcial.*

Valor (*Bits* 27 e 67)	Significado
0	Sem transmissão para recepção parcial
1	Com transmissão para recepção parcial

A modulação utilizada por cada camada de transmissão é indicada nas portadoras TMCC através do esquema de modulação que pode assumir os valores da Tabela 2.16. Nas Tabelas 2.17 e 2.18 são mostrados os valores possíveis de configuração da taxa de codificação e os valores possíveis de configuração do entrelaçador temporal.

2.2.13 A capacidade de transmissão do sistema

A capacidade de transmissão do sistema varia de acordo com as configurações selecionadas. Calcula-se o número de pacotes de 204 *bytes* (após *Reed-Solomon*) transmitidos durante um quadro OFDM através da expressão:

Tabela 2.16: *Esquema de modulação.*

Valor (*bits* 28 a 30/41 a 43 / 54 a 56 68 a 70 / 81 a B83 / 94 a 96)	Significado
000	DQPSK
001	QPSK
010	16QAM
011	64QAM
100-110	Reservado
111	Camada hierárquica não utilizada

Tabela 2.17: *Taxas do código convolucional.*

Valor (*bits* 31 a 33 / 44 a 46 / 57 a 59 71 a 73 / 84 a 86 / 97 a 99)	Significado
000	1/2
001	2/3
010	3/4
011	5/6
100	7/8
101-110	Reservado
111	Camada hierárquica não utilizada

Tabela 2.18: *Entrelaçador temporal.*

Valor (*bits* 34 a 36 / 47 a 49 / 60 a 62 74 a 76 / 87 a 89 / 100 a 102)	Significado
000	0 (Modo 1), 0 (Modo 2) , 0 (Modo 3)
001	4 (Modo 1), 2 (Modo 2) , 1 (Modo 3)
010	8 (Modo 1) , 4 (Modo 2) , 2 (Modo 3)
011	16 (Modo 1) , 8 (Modo 2) , 4 (Modo 3)
100-110	Reservado
111	Camada hierárquica não utilizada

$$ppq = npu \times bpp \times tcd \times \frac{1}{204 \times 8}, \tag{2.5}$$

onde *ppq* representa o número de pacotes transmitidos por quadro OFDM, *npu* o número de portadoras úteis, *bpp* o número de *bits* por portadora e *tcd* é a taxa

Tabela 2.19: *Número de segmentos: possíveis valores e seus significados*

Valor (*bits* 37 a 40 / 50 a 53 / 63 a 66 77 a 80 / 90 a 93 / 103 a 106)	Significado
0000	Reservado
0001	1 segmento
0010	2 segmentos
0011	3 segmentos
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	Reservado
1111	Camada hierárquica não utilizada

de puncionamento. A taxa útil de transmissão é dada por

$$R = ppq \times \frac{188}{204} \times tq, \tag{2.6}$$

sendo que tq é o tempo de duração do quadro OFDM em segundos. A Tabela 2.20 mostra a taxa útil, de acordo com a configuração do sistema, considerando todos os 13 segmentos. Já a Tabela 2.21 mostra o número equivalente de pacotes transmitidos após a codificação *Reed-Solomon*.

2.3 Conclusão

Este capítulo mostra que o subsistema de transmissão de TV digital brasileira possui diversos pontos de configuração, que permitem adequar o sinal de transmissão de acordo com a necessidade da emissora, como taxa útil de transmissão e robustez do sinal irradiado. As informações mostradas ao longo deste capítulo servem de base para o estudo da parte de transmissão e também de multiplexação do sistema ISDB-T_B.

Tabela 2.20: *Taxa útil de dados em 13 segmentos.*

Modulação	Código Conv.	Taxa de dados (Mbps)			
		Tg: 1/4	Tg: 1/8	Tg: 1/16	Tg: 1/32
DQPSK QPSK	1/2	3,651	4,056	4,295	4,425
	2/3	4,868	5,409	5,727	5,900
	3/4	5,476	6,085	6,443	6,638
	5/6	6,085	6,761	7,159	7,376
	7/8	6,389	7,099	7,517	7,744
16QAM	1/2	7,302	8,113	8,590	8,851
	2/3	9,376	10,818	11,454	11,801
	3/4	10,953	12,170	12,886	13,276
	5/6	12,170	13,522	14,318	14,752
	7/8	12,779	14,198	15,034	15,489
64QAM	1/2	10,953	12,170	12,886	13,276
	2/3	14,604	16,227	17,181	17,702
	3/4	16,430	18,255	19,329	19,915
	5/6	18,255	20,284	21,477	22,128
	7/8	19,168	21,298	22,551	23,234

Tabela 2.21: *Número de pacotes de 204 bytes transmitidos nos 13 segmentos.*

Modulação	Código Conv.	Número de pacotes		
		modo 1	modo 2	modo 3
DQPSK QPSK	1/2	156	312	624
	2/3	208	416	832
	3/4	234	468	936
	5/6	260	520	1040
	7/8	273	546	1092
16QAM	1/2	312	624	1248
	2/3	416	832	1664
	3/4	468	936	1872
	5/6	520	1040	2080
	7/8	546	1092	2184
64QAM	1/2	468	936	1872
	2/3	624	1248	2496
	3/4	702	1404	2808
	5/6	780	1560	3120
	7/8	819	1638	3276

Capítulo 3

Camada de multiplexação do padrão ISDB-T_B

Neste capítulo são apresentados detalhes a respeito da formação do TS e do BTS, que são o fluxo de entrada e saída do multiplexador, respectivamente. Este capítulo está estruturado da seguinte maneira: a Seção 3.1 mostra a formação do TS, a Seção 3.2 mostra conceitos relacionados a multiplexação de diferentes TS's, a Seção 3.3 mostra a formação do BTS e a Seção 3.4 encerra o capítulo com uma conclusão.

3.1 Fluxo de dados da camada de transporte do padrão MPEG-2

As informações de áudio e de vídeo a serem transmitidas no sistema de TV Digital passam pelo processo de compressão com objetivo de reduzir a sua taxa de *bits* e, consequentemente, reduzir a largura de banda necessária para sua transmissão. O avanço das técnicas de compressão de áudio e de vídeo foram de suma importância para o desenvolvimento do sistema de TV Digital. Para a transmissão de um sinal de vídeo com definição padrão (*Standard Definition Television* - SDTV) sem nenhuma compressão, por exemplo, seria necessária a taxa de aproximadamente 270 Mbps [16] enquanto que a maior taxa de transmissão disponível no sistema ISDB-T_B não ultrapassa os 24 Mbps [2]. Graças a utilização de técnicas de compressão MPEG-4 é possível reduzir a taxa necessária para transmissão de um sinal SDTV para cerca de 2.5 Mbps sem perdas significativas na qualidade de imagem, conforme testes práticos realizados em laboratório.

Em uma estação de TV, o processo de compressão é realizado por equipamen-

tos chamados codificadores. Os codificadores recebem em sua entrada o sinal de vídeo e de áudio digital sem compressão e entregam em sua saída estes sinais comprimidos e organizados em um fluxo chamado de TS, cuja formação será descrita nas seções seguintes.

3.1.1 Organização dos dados comprimidos

Fluxo Elementar ou *Elementary Stream* (ES) é o nome dado ao fluxo de dados correspondente aos sinais de áudio ou de vídeo logo após o processo de compressão. Após ser organizado em pacotes de tamanho variável, esse fluxo passa a ser chamado de Fluxo Elementar Empacotado ou PES (*Packetized Elementary Stream*)[16],[17]. Os pacotes do PES possuem um cabeçalho inicial seguidos pela sua carga. O cabeçalho transporta, entre outras informações, a identificação dos dados presentes em sua carga (áudio, vídeo ou dados), a indicação do tamanho do pacote PES e as informações de tempo (estampas de tempo) para informar ao decodificador quando decodificar (*Decoding Time Stamps*- DTS) ou apresentar (*Presentation Time Stamps*- PTS) as informações de vídeo e de áudio.

O comprimento do pacote PES está relacionado com seu tempo de exibição (quadro). Como a taxa de compressão varia instantâneamente de acordo com o sinal de áudio e vídeo, o PES possui tamanho variável [16]. Na sequência, o PES é dividido em pacotes menores e de tamanho fixo (188 *bytes*) dando origem ao fluxo chamado de *Transport Stream* (TS). Com a utilização de pacotes de tamanho menor fica mais fácil o controle de erros nos dados bem como a multiplexação de diversos programas em um mesmo fluxo (ex. multi-programação) [16]. O TS é, portanto, um fluxo de dados que transporta principalmente os sinais de áudio e vídeo multiplexados e já comprimidos, além de pacotes de informações adicionais sobre o fluxo. A Figura 3.1 ilustra a formação do TS a partir do PES e a formação do PES a partir do ES.

3.1.2 Formação dos pacotes do TS

O TS é o fluxo da camada de transporte do padrão MPEG-2, formado por pacotes de tamanho fixo de 188 *bytes*, sendo que 4 *bytes* são de cabeçalho e 184 de carga. O TS é formado por pacotes que transportam diferentes tipos de informações, como partes de pacotes PES (cuja compressão de vídeo e áudio utilizada é MPEG-4 e AAC respectivamente, para o caso do sistema ISDB-T_B), informações sobre os serviços, dados e informações de enchimento (pacotes nulos) que são utilizados para obter uma taxa de transmissão fixa. A seguir, está mostrado com maiores detalhes a estrutura de formação dos pacotes do TS.

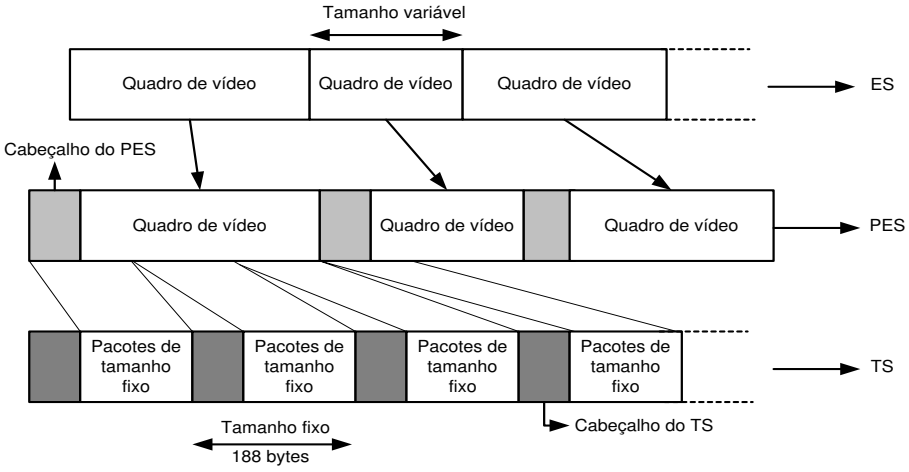


Figura 3.1: *Formação do ES, PES e TS.*

Cabeçalho de cada pacote

Na Figura 3.2 é possível visualizar um pacote do TS, bem como a formação do cabeçalho do mesmo. Das informações contidas no cabeçalho, pode-se destacar o *byte* de sincronismo que indica o início de cada pacote e o PID (*Packet Identifier*) que é a identificação do pacote.

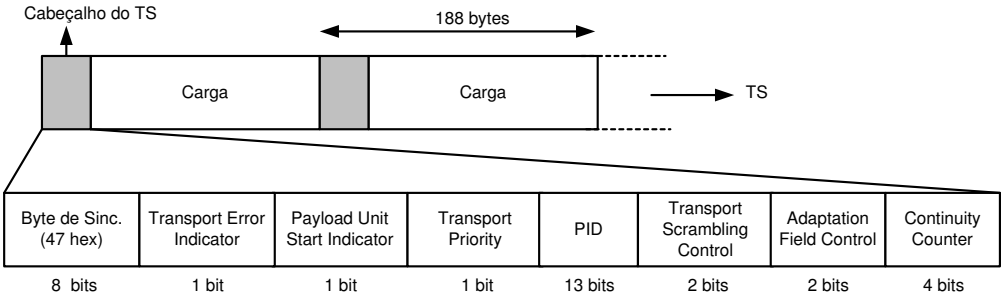


Figura 3.2: *Formação do cabeçalho do TS.*

A seguir há uma descrição de cada campo que compõe o cabeçalho:

- *Byte* de sincronismo: é utilizado como referência do início de um pacote do TS e possui seu valor fixo e igual a 47_h [10]. Esse valor pode estar contido em outras partes do pacote, porém haverá sempre 188 *bytes* entre 2 *bytes* de sincronismo consecutivos [17].
- *Transport error indicator*: é um campo formado por 1 *bit* que indica se o pacote possui erros, ocasionados durante a transmissão, que não puderam ser corrigidos pelos mecanismos de controle de erro existentes no sistema de

transmissão. Esse campo carrega o valor 1 quando detectado pelo menos 1 *bit* errado no pacote e carrega o valor 0 caso contrário [10]. Caso o *bit transport error indicator* indique a presença de um erro, o pacote é descartado pelo decodificador MPEG [16].

- *Payload unit start indicator*: nos pacotes nulos, esse campo assume o valor '0'; para pacotes que transportam tabelas do sistema assume '1' quando os dados de formação da tabela se iniciam naquele pacote e '0' caso contrário. No caso do pacote estar transportando informações de áudio ou de vídeo, assumirá o valor '1' quando o pacote estiver transportanto o início de um pacote PES e '0' caso contrário. Para pacotes que transportem dados, a norma não especifica valores para esse campo [10].

- *Transport priority*: quando esse campo de um *bit* transporta o valor '1', indica que o pacote em questão tem prioridade maior que os demais pacotes do mesmo PID que tenham o campo *Transport Priority* com valor '0'. A norma do MPEG [10] sugere que este recurso pode ser utilizado para priorizar os pacotes que transportem conteúdo de um ES.

- PID - *Packet identifier*: é um campo formado por 13 *bits* que indentifica cada pacote do TS. O PID é atribuído ao pacote de acordo com tipo de informação que ele está transportando em sua carga. Por exemplo, os pacotes que transportam informações de ES diferentes e os pacotes que transportam dados ou tabelas do sistema deverão ter valores de PID diferentes entre si. Alguns valores de PID são reservados ou pré-estabelecidos pela norma do MPEG, tal como mostra a Tabela 3.1, e outros são reservados para a transmissão de tabelas do sistema, conforme mostra a Sessão 3.1.4.

- *Transport scrambling control*: indica se o conteúdo transportado pelo pacote sofreu algum tipo de criptografia. A utilização de criptografia pode restringir o acesso a determinados pacotes a somente um grupo de usuários que tiverem permissão para tal.

- *Adaptation field control*: formado por 2 *bits* que indicam se entre o cabeçalho inicial e a carga do pacote existe um campo especial chamado de Campo de Adaptação ou *adaptation field*. Os valores possíveis para os 2 *bits* e seus respectivos significados estão mostrados na Tabela 3.2.

- *Continuity counter*: É um campo formado por 4 *bits* que transporta o valor de um contador que é incrementado a cada pacote enviado com o mesmo PID. Este contador possibilita identificar a perda ou repetição de pacotes, já que o valor esperado para os pacotes de um mesmo PID deve assumir sequencialmente os valores de 0 a 15. Ao término da contagem, o contador

é reiniciado e descontinuidades nesse contador podem ser indicadas através de um sinalizador especial localizado no *adaptation field*.

Tabela 3.1: *Tabelas de PID's reservados pela norma do MPEG.*

Valor do PID	Descriçao
0x0000	Pacote que transporta a tabela PAT
0x0001	Pacote que transporta a tabela CAT
0x0002 - 0x000F	Valores de PID's reservados
0x1FFF	Identificação do pacote nulo

Tabela 3.2: *Valores possíveis para o adaptation field control.*

Valor do campo	Significado
00	Valor reservado
01	Pacote sem *adaptation field*, somente com *payload*
10	Pacote sem *payload*, com *adaptation field* somente
11	Pacote com *adaptation field* seguido pelo *payload*

Adaptation field

Conforme comentado na Sessão 3.1.2 pode-se transmitir entre o cabeçalho do TS e sua carga um campo especial chamado de *adaptation field*. O quinto *byte* do pacote (*byte* logo após o cabeçalho) é chamado de *adaptation field length* e indica a quantidade de *bytes* logo após o quinto *byte* que serão utilizados para transportar as informações do *adaptation field*. Na Figura 3.3 é possível visualizar um exemplo de um pacote com *adaptation field*.

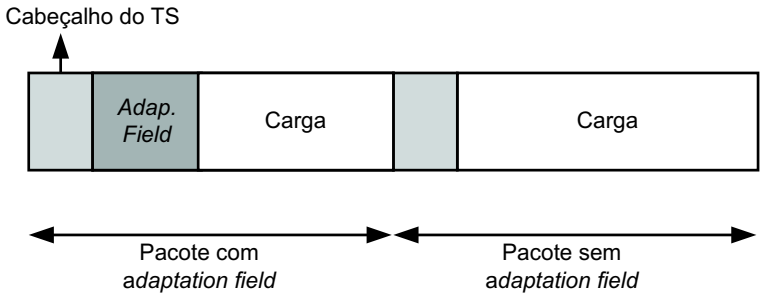


Figura 3.3: *Pacote do TS com adaptation field.*

A existência ou não do *adaptation field* é sinalizada no cabeçalho do TS através do campo *adaptation field control* (ver Tabela 3.2). Quando não existe o campo

adaptation field no pacote, o quinto *byte* deverá indicar comprimento 0. Quando o pacote transportar tanto *adaptation field* quanto dados úteis em sua carga, o *adaptation field length* pode indicar valores entre 0 e 182; caso o pacote transporte somente o adaptation field, o *adaptation field length* deverá transportar o valor 183. Dentre as informações transportadas no *adaptation field* pode-se destacar o *program clock reference* (PCR) que está apresentado com mais detalhes a seguir.

3.1.3 Referências de tempo do TS

A utilização de referências ou estampas de tempo transportadas no TS possibilita a correta e síncrona decodificação e apresentação dos sinais de vídeo e de áudio. Conforme comentado na Sessão 3.1, o PES transporta marcações de tempo que indicam quando o quadro de vídeo ou de áudio deve ser decodificado (DTS) e apresentado (PTS) pelo decodificador. Essas marcações de tempo são geradas baseadas em um *clock* de 27 MHz do codificador. Essas devem ser respeitadas e atendidas no decodificador que as analisa tendo como base um *clock* de mesmo valor. Com o intuito de sincronizar o sistema de *clock* (STC - *System Time Clock*) do codificador com o do decoficador, são enviadas marcações de tempo, chamadas de PCR, nos pacotes do TS. Essas marcações são utilizadas para diminuir as diferenças entre os dois sistemas e fazer com que as contagens de tempo sejam idênticas em ambos. A seguir, estão mostrados maiores detalhes a respeito do PCR.

PCR

O PCR é formado por dois contadores que são incrementados com base no *clock* de 27 MHz. Esses contadores existem tanto no codificador quanto no decodificador. O decodificador recebe e extraí do TS o valor dos contadores gerados pelo codificador e os compara com o valor do seu contador local. A diferença encontrada alimenta o controle de geração do *clock* local de forma a buscar maior sincronismo entre eles. Para que o sistema opere perfeitamente, os valores de PCR devem ser enviados com certa frequência e as diferenças entre os valores dos contadores recebidos pelo decodificador e seu contador local devem ter certa precisão. A norma do MPEG indica que o máximo espaçamento de tempo entre PCR transmitidos seja de 40 ms e que erros de precisão não ultrapassem os +/- 500 ns [10]. O sistema ISDB, por outro lado, admite espaçamento entre pacotes maior que 60 ns [2].

O PCR é formado por dois contadores ou dois sub-campos que são o PCR *Base*, formado por 33 *bits* e o PCR *Extension* formado por 9 *bits*. O último transporta o valor de um contador que é incrementado a cada pulso de *clock* do

STC (27 MHz), até que seja atingido o valor 299, quando então tem sua contagem reinicializada a partir do 0. A cada reinicialização desse contador, o valor do PCR *Base* é incrementado de 1. Portando, podemos dizer que o PCR *Extension* tem como referência de contagem o próprio valor de *clock* do STC, enquanto o PCR *Base* tem como referência um *clock* trezentas vezes menor, ou seja, 90 kHz. A partir dos valores de dois PCR's consecutivos é possível determinar o tempo que se passou entre a geração deles e contabilizando-se a quantidade de *bytes* entre eles, consegue-se determinar a taxa de dados do fluxo, conforme será explicado a seguir.

A Figura 3.4 mostra um pacote que transporta o PCR de um TS capturado da saída de um codificador. Na parte inferior direita da figura está o pacote de 188 bytes e na parte superior direita está mostrada a análise dos dados transportados no pacote. O *software* utilizado para visualizar o conteúdo do pacote é o MPEG TS *Compliance Analyzer*, desenvolvido pela empresa Tektronix [18]. Este *software* realiza análise de conformidade da formação do TS incluindo análises de PCR, conformidades da formação das tabelas e análises relacionadas ao padrão ISDB-T. A Tabela 3.3 mostra as informações retiradas de 4 pacotes consecutivos que transportam informações sobre o PCR.

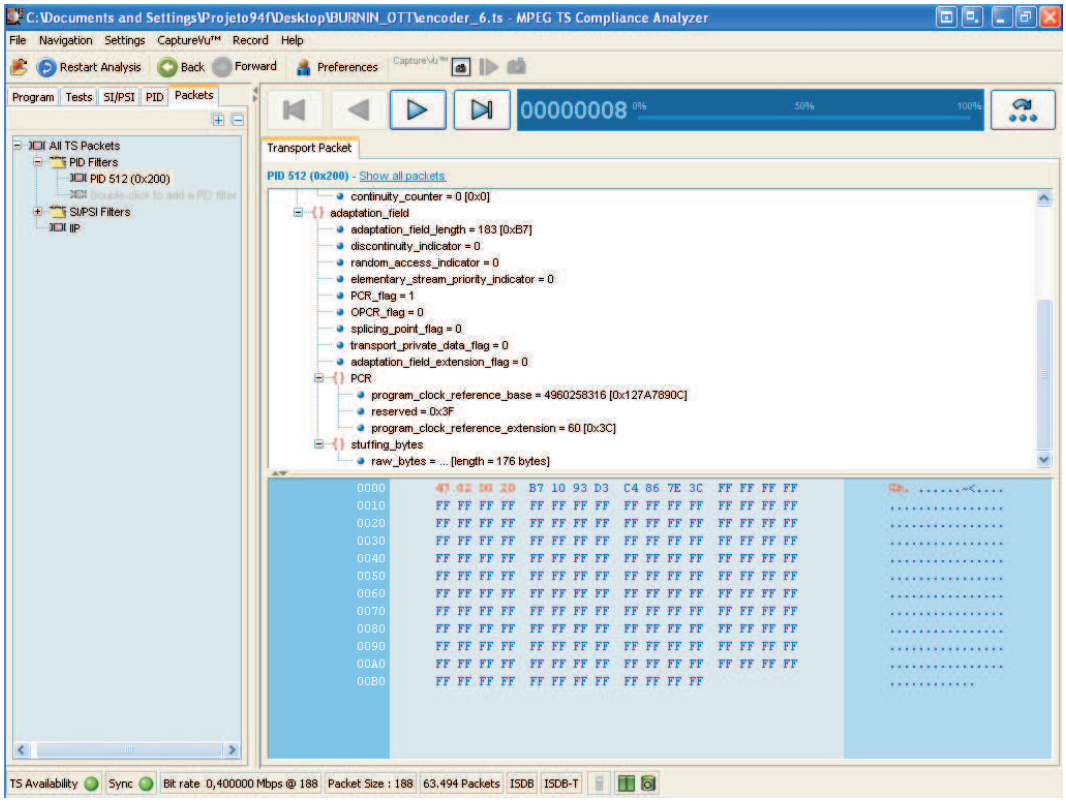


Figura 3.4: *Pacote transportando o PCR.*

O valor do PCR para um determinado pacote, com índice *ip*, é dado por:

Tabela 3.3: *Valores de PCR extraídos de um TS.*

Índice do pacote no TS	PCR *Base*	PCR *Extension*	Valor calculado do PCR
8	4960258316	60	1488077494860
17	4960261361	240	1488078408540
26	4960264407	120	1488079322220
35	4960267453	0	1488080235900

$$PCR(ip) = (PCR_b \times 300) + PCR_e \tag{3.1}$$

onde PCR_b respresenta o valor do campo PCR *Base* e PCR_e o PCR *Extension*. O número de pulsos de *clock* de 27 MHz entre dois pacotes consecutivos com PCR, *pep*, pode ser calculado por

$$pep = pcr(ip'') - pcr(ip'), \tag{3.2}$$

sendo $pcr(ip'')$ o valor do PCR extraído de um dado pacote e $pcr(ip')$ o valor do PCR extraído do pacote anterior. A partir da quantidade de pulsos de *clock* pode-se calcular o intervalo de tempo existente entre os pacotes consecutivos que é dado por:

$$tep = \frac{pep}{27000000} \ \texttt{s}. \tag{3.3}$$

A partir do índice *ip*, que indica a posição do pacote no TS, é possível calcular a quantidade de dados transmitidos (btx) entre os dois pacotes consecutivos admitindo que

$$btx = (ip'' - ip') \times 188 \times 8 \ \texttt{bits}. \tag{3.4}$$

E, por fim, a taxa do fluxo (Rc) é dada por:

$$Rc = \frac{btx}{tep} \ \texttt{bps}. \tag{3.5}$$

A partir dos dados extraídos dos pacotes de índice 17 (ip'') e 8 (ip') da Tabela 3.3 e utilizando (3.3), (3.4) e (3.5) pode-se calcular que há 913.680 pulsos de *clock* entre ambos os pacotes, cujo espaçamento temporal calculado é de 0,0338400 s. A quantidade de *bits* transmitidos nesse intervalo de tempo é de 13536, resultando em uma taxa de transmissão do fluxo de 400 kbps. Esse valor calculado coincide com a medida pelo *software* MPEG TS *Compliance Analyzer* (ver campo *Bit rate* na parte inferior da Figura 3.4).

Pode-se concluir que o PCR é uma importante referência para a decodificação do TS e que, para um dado fluxo, tanto a distribuição dos pacotes que o formam quanto sua taxa de transmissão, estão diretamente relacionadas com os valores transportados pelo PCR.

3.1.4 Tabelas do sistema

Conforme comentado na Sessão 3.1.2, cada pacote que forma o TS possui uma identificação individual, que é o PID. No entanto, não existe um valor de PID pré-determinado para o vídeo ou áudio. Alguns pacotes transportam informações adicionais sobre o TS e informam como o fluxo está estruturado, identificando quantos programas estão sendo transmitidos, qual áudio e qual vídeo pertencem a um determinado programa, o que cada PID transporta, entre outras informações. Essas informações sobre o fluxo são chamadas de tabelas. Elas possuem uma estrutura de formação muito bem definida e são transportadas na carga de pacotes. Cada tabela possui um valor de PID associado, conforme mostra a tabela 3.4 [7].

Tabela 3.4: *PID's utilizados para transmissão de tabelas do sistema.*

Valor do PID	Tabela associada
0x0000	PAT (*Program Association Table*)
0x0001	CAT (*Conditional Access Table*)
0x0010	NIT (*Network Information Table*)
0x0011	SDT (*Service Description Table*) e BAT (*Bouquet Association Table*)
0x0012, 0x0026 e 0x0027	EIT (*Event Information Table*)
0x0013	RST (*Running Status Table*)
0x0014	TDT (*Time and Date Table*) e TOT (*Time Offset Table*)
0x0022	PCAT (*Partial Content Announcement Table*)
0x0024	BIT (*Broadcaster Information Table*)
0x0025	NBIT (*Network Board Information Table*) e LDT (*Linked Description Table*)

A Seção 3.1.5 descreve a finalidade das tabelas aqui listadas.

3.1.5 As tabelas SI/PSI

As tabelas SI/PSI (*Service Information / Program Specific Information*) são um conjunto de tabelas que permitem ao receptor identificar e decodificar corretamente os programas que formam um TS. A tabela PAT, por exemplo, associa cada programa do fluxo ao PID dos pacotes que transportam a sua PMT. A

PMT, por sua vez, é uma tabela que lista os PIDS dos pacotes de áudio, vídeo, dados, PCR, entre outros, que estão associados a um determinado programa. Na Figura 3.5 está mostrado um exemplo da estrutura de um TS baseada nas informações retiradas dessas duas tabelas. Nesse exemplo, a PAT informa que existem dois programas no fluxo e associa a cada um deles um valor de PID da tabela PMT. A tabela PMT, por sua vez, informa os PID'S dos pacotes associados ao programa em questão.

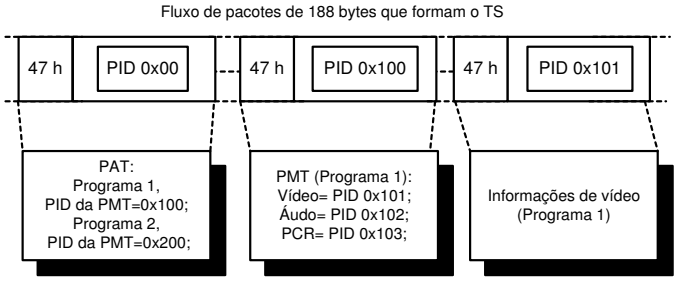


Figura 3.5: *Estrutura de um TS definida pelas tabelas PAT e PMT.*

As tabelas são transmitidas com uma frequência relativamente alta para que seja possível que um receptor ligado em um instante qualquer possa rapidamente identificar e decodificar os dados recebidos. As tabelas PAT e PMT, por exemplo, devem ser enviadas em média a cada 100 ms [7].

As tabelas podem sofrer alterações a qualquer instante e tais alterações são informadas através de um controle de versão (*Version Number*) da tabela e indicação de qual versão da tabela deve ser aplicada (*Current Next Indicator*) no momento.

As tabelas são formadas de acordo com uma estrutura chamada de Seção. A Seção tem tamanho variável e é transportada dentro da carga de um pacote do TS. Para que o receptor possa identificar o tamanho da tabela corretamente, é informado em quantas Seções a tabela foi dividida (*Last Section Number*), além da identificação da Seção transportada em cada pacote (*Section Number*). Ao término da tabela, são acrescentados 4 *bytes* de CRC(*Cyclic Redundancy Check*) que possibilitam a identificação de erros no receptor.

As principais tabelas PSI e SI e suas funções estão mostradas nas tabelas 3.5 e 3.6, respectivamente.

A seguir, apresenta-se uma descrição detalhada a respeito da formação e função das tabelas mais importantes para a correta decodificação e exibição de um TS, que são as tabelas PAT, PMT, NIT e CAT.

Tabela 3.5: *Principais Tabelas PSI definidas pelo MPEG.*

Nome	Descrição / Função
PAT	Tabela de associação de programas. Associa cada programa à um PID de PMT.
PMT	Tabela de mapeamento de programas. Especifica os componentes de um programa ou serviço.
CAT	Tabela de acesso condicional. Informações sobre Acesso Condicional aos programas.

Tabela 3.6: *Principais Tabelas SI utilizadas no sistema ISDB-T_B.*

Nome	Significado / Descrição
BIT	Tabela de informação do radiodifusor. Designa as unidades radiodifusoras e os parâmetros SI para cada uma delas.
EIT	Tabela de informação de eventos. Transporta informações sobre os eventos existentes em cada serviço bem como sua ordem cronológica de exibição.
NIT	Tabela de informação de rede. Transporta informações sobre a formação do TS e sobre o seu meio de transmissão.
RST	Tabela de estado do evento. Permite atualização do estado de eventos, por exemplo, quando há alteração do horário de programação (ex. pausa, rodando).
SDT	Tabela de descrição de serviços. Lista e identifica os serviços existentes em um TS.
TDT	Tabela de data e horário. Informa hora e data do sistema.
TOT	Tabela de mudança de data e horário. Informa ao receptor a hora, data e fuso horário.

PAT - Tabela de Associação de Programas

Para que um receptor de TV Digital possa decodificar um programa presente em um fluxo recebido é necessário identificar que programas estão contidos nele. A tabela PAT informa ao receptor os programas presentes no fluxo, identificado-os através de um número chamado número do programa (*Program Number*), e os associa ao valor do PID dos pacotes que transportam a tabela PMT associada. Na Figura 3.6 está mostrada a estrutura de formação da PAT. Na sequência, há uma descrição dos campos que a formam. Será utilizada a sigla RSV para indicar

os campos reservados para uso futuro.

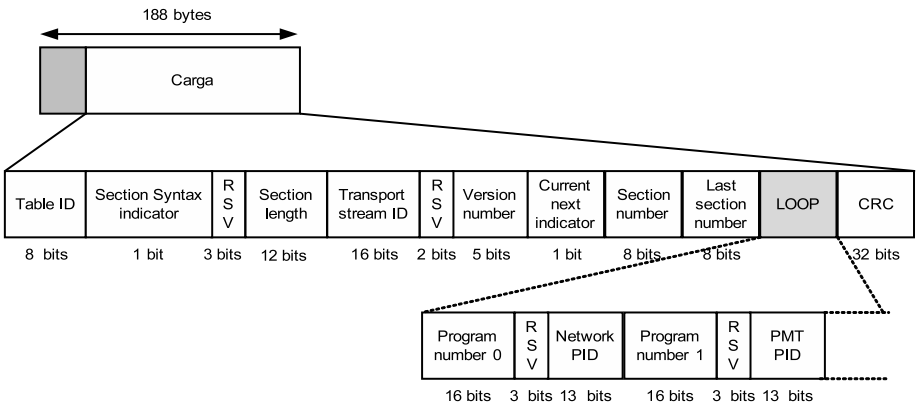


Figura 3.6: *Estrutura de formação da PAT.*

- *Table ID*: identifica a tabela que está sendo transportada na seção. No caso da PAT o *Table* ID deve assumir o valor $0x00_h$.
- *Section Length*: o comprimento em *bytes* que a seção ocupa, que não possui um tamanho pré-determinado, é indicado através deste campo. Ele indica a quantidade de *bytes* da seção contando a partir do *Section Length* e incluindo os *bytes* de CRC [10].
- *Transport Stream ID*: é um valor definido pelo usuário que identifica o TS.
- *Version Number*: a cada nova atualização na tabela, é incrementado o valor do campo *Version Number*. Mas a tabela só passa a ser válida quando houver sinalização no campo *Current Next Indicator*.
- *Current Next Indicator*: sinaliza se a versão de tabela transportada no pacote atual é válida ou não. Quando esse campo carrega o valor '0', indica que a versão da tabela em questão ainda não deve ser aplicada. Caso esse campo transporte o valor '1', indica que a versão da tabela atual é a válida.
- *Section Number*: pode ser necessário mais de uma seção para transportar uma tabela. No caso, o campo *Section Number* indica qual o número da seção atual, sendo que a primeira seção tem *Section Number* igual a '0'.
- *Last Section Number*: este campo indica qual é o valor do *Section Number* da última seção que contém a tabela.

- *CRC*: 4 *bytes* de paridade que são adicionados ao final da seção para possibilitar a identificação de erros pelo receptor.

A maioria dos campos descritos anteriormente são comuns às demais tabelas SI/PSI, com exceção do *Transport Stream* ID, pois fazem parte da formação de uma seção. A seguir estão listados os campos que são exclusivos da PAT.

- *Program Number*: é o número que identifica cada serviço (programa) do fluxo. O papel da PAT é associar cada *Program Number* ao valor de PID dos pacotes que transportam sua correspondente PMT. Esse número pode ser configurado pelo usuário com exceção do *Program Number* de valor 0x00 que é reservado para indicação da NIT.

- *Program Map* PID (PMT PID): Número do PID do pacote que transporta a PMT do programa identificado pelo *Program Number*.

PMT - Tabela de Mapeamento de Programas

Através das informações transportadas pela PMT é possível que o receptor identifique qual áudio, vídeo e dados devem ser decodificados para cada serviço do fluxo recebido, bem como qual referência de tempo (PCR) deve ser utilizada. A PMT é, portanto, a responsável por mostrar ao receptor como está estruturado cada serviço. Deve existir uma tabela PMT para cada programa do fluxo.

Está mostrado na Figura 3.7 como é estruturada a tabela PMT. Parte dos campos que a formam são idênticos a campos encontrados na PAT. No caso da PMT, o *Table ID* deve assumir o valor $0x02_h$. A PMT identifica o programa (*Program Number*) e o associa a um PCR (PCR PID). Informações de tamanho variável cujo comprimento é informado em *Program Info Length* sobre o programa (*N Loop Descriptors*) também são transmitidos pela PMT. Ela lista todos os elementos relacionados ao programa em questão (áudio, vídeo, dados, etc) e para cada um desses lista 3 informações: qual o tipo do fluxo (*Stream Type*), seu PID (*Elementary PID*) e descritores adicionais (*ES info Length*).

Os campos exclusivos da PMT estão descritos a seguir:

- *Program Number*: identifica qual o programa está associado à PMT. Deve estar em concordância com o informado na PAT.

- PCR PID: identifica em qual PID está sendo transportado o PCR relacionado ao programa em questão. Na prática, vê-se normalmente o PCR sendo enviado em PID's exclusivos ou no mesmo PID utilizado para o transporte do vídeo ou áudio do programa.

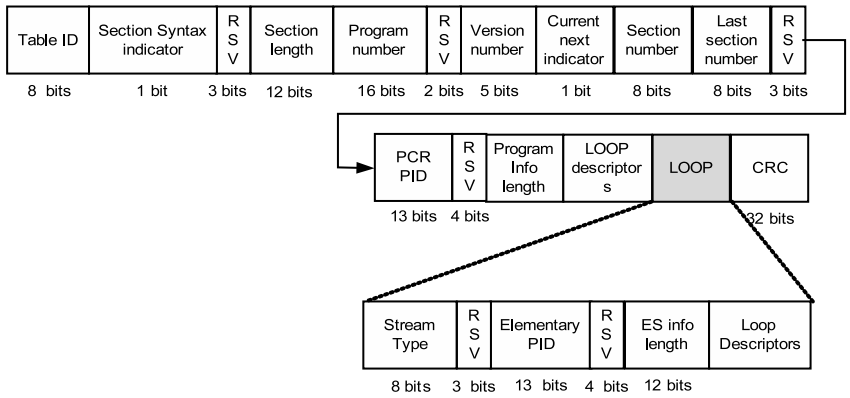


Figura 3.7: *Estrutura de formação da PMT.*

- *Program Info Length*: identifica a quantidade de *bytes* que seguem que são associados a informações do programa.
- *Stream Type*: identifica o tipo de dado que está sendo transportado pelo PID especificado no *Elementary PID*. Alguns dos tipos possíveis são áudio, vídeo, dados de carrossel, pacote PES, etc. [8].
- *Elementary* PID: indica o valor do PID dos pacotes que transporta o elemento do programa.
- *ES info Length*: indica a quantidade de *bytes* que transportam descritores relacionados ao elemento do programa.

Os descritores são informações adicionais que podem ser incluídas na tabela. Vários são os descritores que podem ser associados à PMT como: descritor de acesso condicional, descritor de direitos de cópia, descritor de classificação indicativa, descritor de informação de emergência, descritor de componente de dados, entre outros [7].

CAT - Tabela de Acesso Condicional

O acesso a determinados programas ou informações transportadas em um TS pode ser controlada ou restringida. Alguns pacotes do TS podem ser transmitidos criptografados, fazendo com que o acesso ao conteúdo destes pacotes sejam restritos a um grupo de usuários que possuam autorização para tal, como por exemplo no caso de TV por assinatura. Esse controle de acesso é realizado pela tabela CAT em conjunto com o sinalizador denominado de *Transport Scrambling control* do cabeçalho dos pacotes. A CAT transmite especificamente ao receptor

informações de como desencriptar os dados (ECM - *Entitlement Control Message* e EMMs - *Entitlement Management Message*) [16]. No caso da transmissão em rede aberta, como é o caso da realizada no Brasil com o sistema ISDB-T_B, não se deseja restringir o acesso da população aos dados transmitidos. Por essa razão a CAT perde um pouco sua importância e por esse motivo não serão abordados maiores detalhes a seu respeito, além da sua estrutura de formação mostrada na Figura 3.8. O *Table ID* utilizado pela CAT é $0x01_h$.

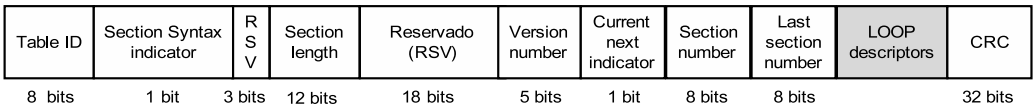


Figura 3.8: *Estrutura de formação da CAT.*

NIT - Tabela de Informação de Rede

A NIT descreve as informações sobre a transmissão do TS informando, por exemplo, se a transmissão do mesmo é via terrestre ou satélite, qual a frequência, quais os parâmetros de modulação utilizados, identificação da emissora entre outros. Essa tabela é considerada uma tabela privada [10] e transporta parâmetros específicos do sistema ISDB-T_B. Na Figura 3.9 está mostrada a estrutura de formação da NIT. Ela possui alguns campos em sua formação que são idênticos aqueles encontrados na formação da tabela PAT. O *Table ID* deve possuir o valor $0x40_h$.

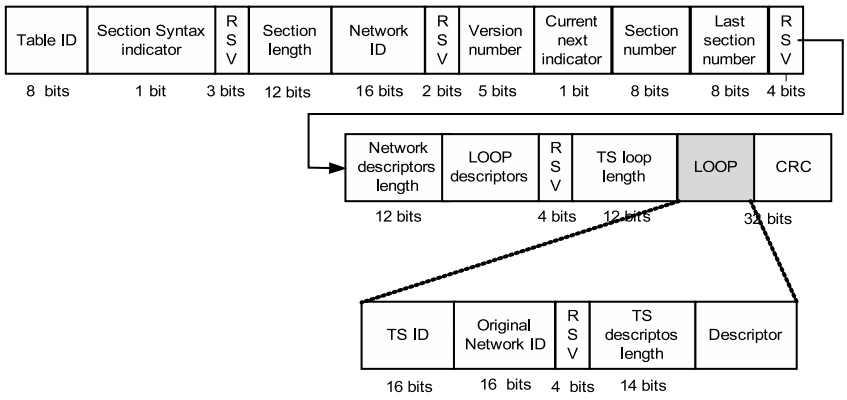


Figura 3.9: *Estrutura de formação da NIT.*

A seguir, apresenta-se a descrição de alguns parâmetros específicos da NIT:

- *Network* ID: identifica a emissora geradora do sinal. O valor que deve ser atribuído a esse campo é exclusivo para cada emissora e é gerado a partir do prefixo concedido pela ANATEL [7].

- *Network Descriptors Length*: indica a quantidade de *bytes* que seguem que são associados a informações da rede de transmissão do TS.

- *Transport Stream Loop Length*: especifica o tamanho total, em *bytes*, do laço (*loop*) de TS, que deve obrigatoriamente terminar imediatamente antes do primeiro *byte* de CRC;

- *Transport Stream* ID: serve como um rótulo para identifica o TS.

- *Original Network* ID: oferece uma identificação única para cada estação geradora e tem mesmo valor do campo *Network* ID.

- *Transport Descriptors Length*: indica a quantidade de *bytes* que seguem que são associados a informações do TS.

Assim como a PMT, a NIT pode transportar vários descritores como, por exemplo, o descritor de sistema de transmissão terrestre (*Terrestrial system delivery descriptor*) que informa o código da área de cobertura do sinal (*Area Code*). Esse código de área tem seu valor gerado com base nas regiões brasileiras [7]. Informa também os parâmetros do sistema de transmissão como frequência central do canal, intervalo de guarda, etc. [7]. Outro descritor é o de recepção parcial (*partial reception descriptor*) que lista os programas do fluxo, identificando-os através do *Program Number*, que estão sendo transmitidos no segmento central do espectro. Já o descritor de informação do TS (*TS information descriptor*) indica em qual camada de transmissão do sistema ISDB (camada A, B ou C) cada um dos programas do fluxo está sendo transmitido. Indica ainda qual a modulação utilizada em cada camada de transmissão (DQPSK, QPSK, 16QAM ou 64QAM) [7]. Além disso, este descritor é o responsável por transportar o canal virtual (*Remote Control Key ID*), que serve para associar um número ao canal digital da emissora. Normalmente, esse campo tem o mesmo valor do número de seu canal analógico, para evitar que o telespectador tenha de memorizar dois números de canais para a mesma emissora: um valor para o canal digital e outro para o canal analógico.

3.1.6 Tabelas adicionais

As tabelas PAT e PMT são tabelas essenciais para a correta decodificação de um TS. As tabelas CAT e NIT também são de grande importância no sistema e

por isso são tabelas de uso obrigatório [7]. A seguir, algumas outras tabelas que também são obrigatórias no sistema ISDB-T_B são apresentadas.

SDT - Tabela de Descrição de Serviços

A tabela SDT lista os serviços existentes no TS e informa, para cada um, o tipo de EPG (*Electronic Programm Guide*) vinculado (EIT-P/F ou EIT-Schedule) além de carregar um rótulo de cada programa e um rótulo para a emissora. Esses rótulos são normalmente exibidos no receptor. Por exemplo, para o da nome da emissora aparecer na tela como "Emissora A"e para nome dos serviços aparecer "Emissora A HD", "Emissora A SD", etc. Essa tabela é transmitida nos pacotes com PID $0x0011_h$.

TOT - Tabela de Mudança de data e horário

A TOT informa ao receptor a data e hora (informando também o fuso horário) e é utilizada como referência para a correta exibição do EPG. Deve ser transmitida pelo menos a cada 30 segundos e utiliza o PID $0x0014_h$.

EIT - Tabela de informação de eventos

A EIT lista em ordem cronológica os chamados eventos contidos em cada um dos serviços. Em outras palavras, lista a grade de programação da emissora. Por exemplo, 12:00h Início do Programa Esportivo, 12:30h Jornal, etc. No exemplo, o programa Esportivo seria um evento e o Jornal seria o próximo evento. São dois os principais tipos de EIT:

- EIT P/F (*present/following*) que indica qual é o evento atual que está sendo exibido pela emissora e qual é o próximo evento a ser exibido. No exemplo dado, se a TOT estiver informando que atualmente são 12:20h, no receptor será exibido que o evento atual é o Programa Esportivo. Para o próximo programa será indicado Jornal.

- EIT SCHEDULE que indica a lista completa de eventos da emissora. Em outras palavras, lista a grade de programação completa, inclusive para várias semanas.

Para cada tipo de receptor existe um tipo de EIT, como mostrado na Tabela 3.7.

Tabela 3.7: *Tipos de receptores e as EIT's associadas.*

Tipo de EIT	PID	Descrição
H-EIT	0x0012	EIT para receptores fixos
M-EIT	0x0026	EIT para receptores móveis
L-EIT	0x0027	EIT para receptores portáteis

Tabelas obrigatórias e demais tabelas

A Tabela 3.8 lista o intervalo de transmissão das tabelas obrigatórias do sistema ISDB-T_B [7]. Embora obrigatórias, as tabelas TOT, EIT e SDT não são essenciais para a correta decodificação de um TS.

Tabela 3.8: *Tabelas obrigatórias.*

Tabela	Intervalo de transmissão
PAT	1 ou mais / 100 ms
CAT	1 ou mais / 1 s
PMT	1 ou mais / 100 ms
NIT	1 ou mais / 10 s
SDT	1 ou mais / 2 s
EIT	1 ou mais / 2 s
TOT	1 ou mais / 30 s

Outras tabelas além das apresentadas podem ser utilizadas no sistema. Algumas, cuja aplicação tem características importantes, estão mostradas na Tabela 3.9.

Tabela 3.9: *Tabelas com aplicações importantes.*

Nome	Significado / Descrição
SDTT	Tabela de início de atualização de *software* de receptores. Transmite informações a respeito da atualização de *software* de receptores, informando hora de atualização e especificados os receptores a serem atualizados.
CDT	Tabelas de dados comuns. Transmite dados como o logotipo da emissora.
AIT	Tabela de informação de aplicação. Transmite informações e controle sobre as aplicações transmitidas no fluxo.

3.2 Multiplexação de TS MPEG-2

O processo de multiplexação de TS acontece quando se deseja unir as informações contidas em dois ou mais fluxos diferentes. Conforme comentado na Sessão 3.1.1, o fato de se utilizar pacotes relativamente pequenos para o transporte das informações no TS facilita o processo de multiplexação. Ao se multiplexar diferentes fluxos, deve-se garantir que a ordem de envio dos pacotes de cada TS não seja alterada, conforme exemplificado pela Figura 3.10. Caso contrário, as informações do TS provavelmente não serão corretamente decodificadas e poderão ocorrer erros de continuidade, indicados pelo *continuit counter*. Um outro ponto importante é que ao se multiplexar diferentes TS, tem de se garantir que as tabelas do novo fluxo formado indiquem corretamente todos os programas e componentes resultantes do processo de multiplexação. Outro ponto é a atualização de PCR que se faz necessária devido ao fato do espaçamento entre pacotes que transportam PCR ser alterado e por, comumente, ser alterada a taxa de transmissão do TS.

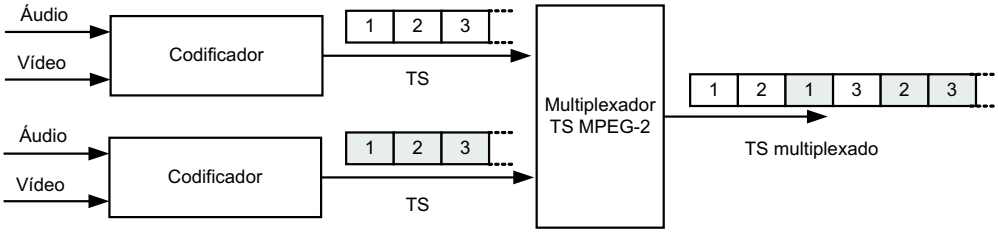


Figura 3.10: *Multiplexação de dois TS's.*

3.3 Geração do BTS

O BTS é o nome dado ao fluxo de saída do multiplexador do sistema ISDB-T_B. O multiplexador deve realizar, além da multiplexação de todos os fluxos com os programas que a emissora deseja transmitir, transformar o TS resultante da multiplexação em um BTS. A seguir, algumas características que diferenciam o BTS de um TS:

- Tamanho dos pacotes: o BTS é formado por pacotes de 204 *bytes*, resultantes do acréscimo de 16 *bytes* ao final de cada pacote do TS.
- Taxa de transmissão: a taxa de dados do BTS é fixa em aproximadamente 32,508 Mbps, ao contrário da taxa do TS que não tem um valor pré-determinado.
- Configuração do transmissor: o BTS transporta informações de configuração do transmissor em um pacote especial chamado de IIP.

- Multiplexação dos pacotes nas camadas hierárquicas do transmissor: a distribuição dos pacotes no BTS deve seguir uma distribuição determinada pela lei de formação do quadro de multiplexação (*Multiplex Frame*).

O quadro de multiplexação possui um período de duração no tempo idêntico ao tempo de duração de 1 quadro OFDM. Ele determina a ordem que os pacotes devem ser entregues às camadas hierárquicas do transmissor. Essa organização leva em consideração a sequência dos dados de cada camada na saída de um receptor modelo, considerando que não há tempo de atraso na demodulação dos dados de cada camada hierárquica [2]. A taxa do quadro de multiplexação é fixa e de valor igual a quatro vezes a taxa de amostragem da IFFT (que irá gerar o sinal OFDM) que é dada por

$$T_q = 4 \times \frac{512}{63} = 32,5079365 \ \texttt{Mbps}. \tag{3.6}$$

Durante a geração do quadro de multiplexação são inseridos pacotes nulos para que essa taxa seja mantida fixa, porém tais pacotes são descartados pelo transmissor.

3.3.1 Número de pacotes por quadro de multiplexação

A quantidade de pacotes que formam um quadro de multiplexação é dado por

$$n_{pq} = t_q \times (\frac{512}{63} \times 10^6 \times 4) \times \frac{1}{204 \times 8} \tag{3.7}$$

onde n_{pq} representa o número de pacotes de 204 *bytes* transmitidos por quadro de multiplexação e t_q é o tempo de duração do quadro OFDM em segundos. Na Tabela 3.10 estão mostrados os números de pacotes por quadro de multiplexação para as possíveis configurações de modo e intervalo de guarda. A Figura 3.11 ilustra um exemplo de um quadro de multiplexação para o modo 1 e intervalo de guarda de 1/8.

Tabela 3.10: *Número de pacotes de 204 bytes por quadro de multiplexação.*

Modo de transmissão	Intervalo de de Guarda 1/4	Intervalo de Guarda 1/8	Intervalo de Guarda 1/16	Intervalo de Guarda 1/32
1	1280	1152	1088	1056
2	2560	2304	2176	2112
3	5120	4608	4352	4224

Comparando a quantidade de pacote por quadro de multiplexação mostrado na Tabela 3.10 com a quantidade de pacotes úteis transmitidos por quadro OFDM

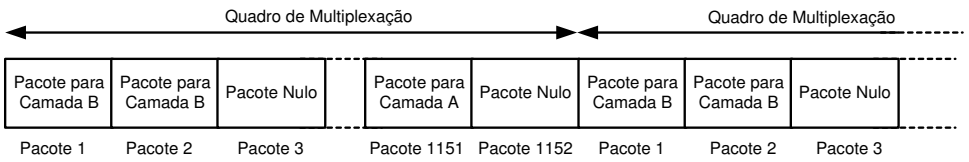


Figura 3.11: *Exemplo de formação do quadro de multiplexação.*

mostrada na Sessão 2.2.13, pode-se concluir que o quadro de multiplexação sempre entregará pacotes além do necessário. Os pacotes inseridos apenas para completar a taxa do quadro de multiplexação são descartados pelo transmissor.

3.3.2 Organização dos pacotes no quadro de multiplexação

Os pacotes que formam o BTS são organizados de acordo com o modelo do receptor do sistema ISDB-T, que está mostrado na Figura 3.12 [2]. Os dados recebidos e demodulados são agrupados de acordo com a camada hierárquica à que pertencem. Em seguida, passam pelo processo de depuncionamento, que consiste na inserção de zeros nos pontos onde os *bits* que foram descartados no processo de puncionamento e são encaminhados ao *buffer* da camada hierárquica (*Buffer* A, *Buffer* B e *Buffer* C na Figura 3.12). O *buffer* da camada hierárquica transfere o conteúdo que armazenou ao *buffer* de TS, através das chaves SW_1 e SW_3, sempre em pacotes de 408 *bytes*. O tamanho dos pacotes se deve ao fato de que 408 *bytes* serão convertidos em 204 *bytes* após o decodificador de Viterbi, já que o código convolucional utilizado tem taxa 1/2. A cada 408 pulsos de *clock* a chave SW_4 deve enviar um pacote de 408 *bytes* ao decodificador de Viterbi. Caso nenhum dos *buffers* de TS tenham dados suficiente para ser repassado ao decodificador, um pacote nulo é gerado e repassado ao decodificador através da chave SW_2. A chave SW_4 só passa a funcionar após 3×408 pulsos de *clock* depois do sistema ter sido inicializado, para garantir que o *buffer* de TS tenha dados armazenados.

A quantidade de *bits*, B_{xk}, que cada camada hierárquica está descarregando em seu seu *buffer* correspondente a cada instante de tempo é dada por

$$B_{xk} = 2 \times ([k \times S_x \times R_x] - [(k-1) \times S_x \times P_x]), \tag{3.8}$$

onde x representa a camada hierárquica (A, B ou C), k representa o número da portadora demodulada, S representa o número de *bits* por símbolo da modulação utilizada pela camada hierárquica e P representa a taxa de puncionamento utilizada. Esse cálculo considera que não há diferença de atraso na demodulação das diferentes camadas.

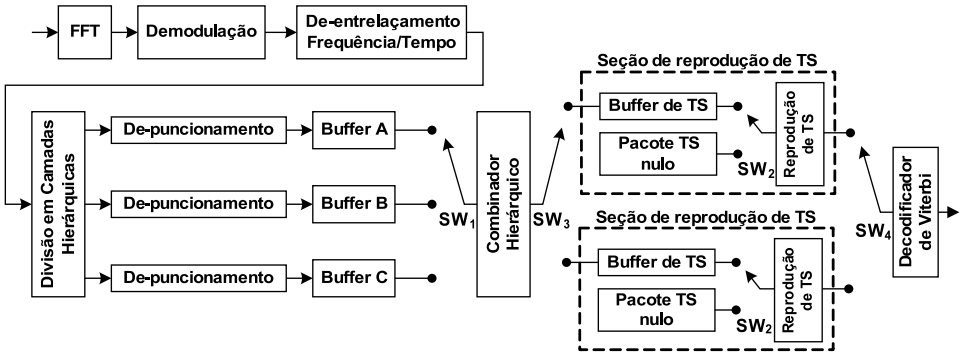


Figura 3.12: *O receptor modelo do sistema ISDB.*

Simulação do modelo do receptor

Uma simulação do modelo do receptor foi desenvolvida, através da qual é possível determinar a formação do quadro de multiplexação. A Figura 3.13 mostra a simulação do modelo do receptor implementada em MATLAB [19]. Na parte superior da simulação configura-se os parâmetros de transmissão e na parte inferior é listada a camada hierárquica de origem dos pacotes obtidos na saída do receptor modelo, sendo que N indica que é um pacote nulo gerado para completar a taxa do quadro. Esta sequência indica a formação e organização do quadro de multiplexação para a configuração selecionada [9].

3.3.3 *Bytes* adicionais nos pacotes do BTS

Na formação do BTS são adicionados 16 *bytes* ao final de cada pacote do TS, conforme mostrado na Figura 3.14. Destes, 8 são informações de controle para o transmissor como, por exemplo, indicação da camada hierárquica na qual o pacote deve ser transmitido. Os outros 8 *bytes* estão previstos para paridade, que é opcional e podem auxiliar na correção de erros inseridos entre a saída do multiplexador e a entrada do transmissor. Caso não sejam utilizados, esses *bytes* normalmente assumem o valor 0xFF_h.

A Tabela 3.11 mostra cada um dos campos que formam os 8 *bytes* de informações de controle [2]. Os 8 *bytes* de paridade, que são de uso opcional, são gerados por um código *Reed-Solomon* encurtado, o RS(204, 196) [2].

3.3.4 IIP

O IIP é um pacote que transporta informações a respeito dos parâmetros de transmissão do sistema ISDB e tem como PID o valor 0x1FC0_h. A cada quadro de multiplexação deve ser transmitido um pacote contendo o IIP. Ele transporta

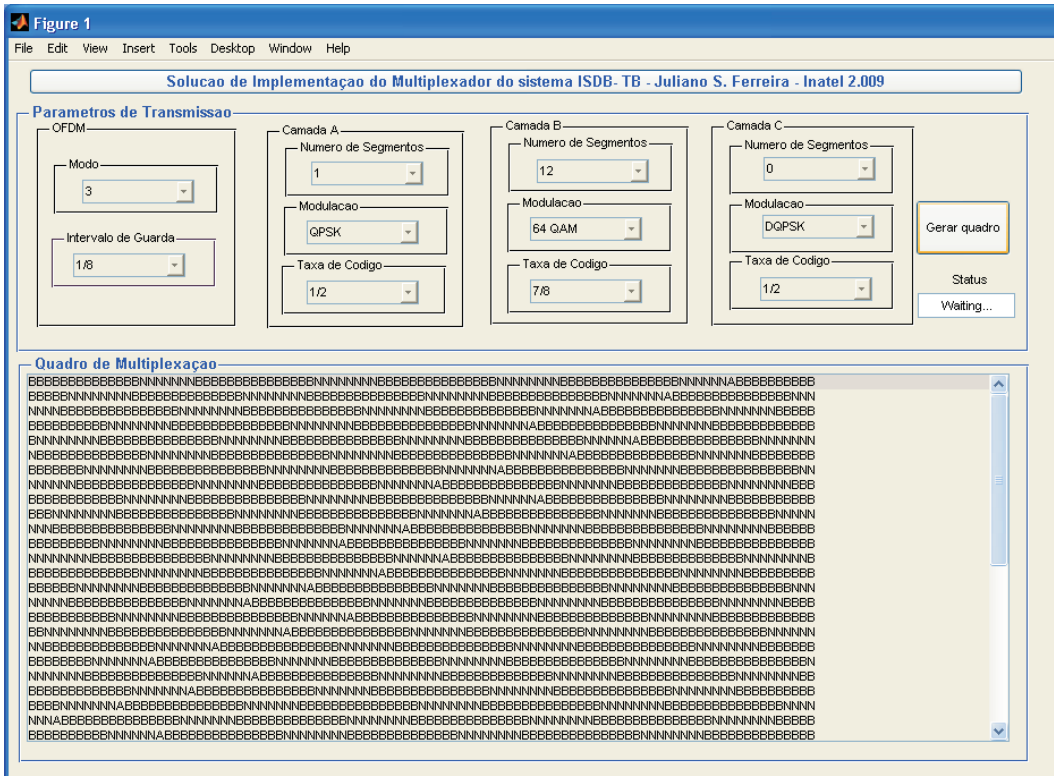


Figura 3.13: *Gerador, em MATLAB, do quadro de multiplexação.*

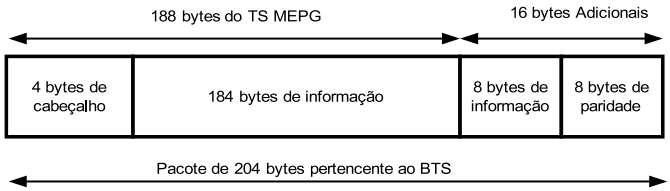


Figura 3.14: *Adição de 16 bytes ao pacote do TS.*

as informações da configuração do transmissor e, em caso de troca de alguns desses parâmetros, controla o processo de transição das configurações, que leva 15 quadros de multiplexação. A Figura 3.15 mostra como estão estruturadas as informações dentro do IIP e, na sequência, tem-se uma descrição dos campos que a formam.

- IIP *Packet Pointer*: indica o número de pacotes, na sequência, para o início do próximo quadro de multiplexação.
- *Modulation Control Configuration Information*: transporta parâmetros de transmissão do sistema. Maiores detalhes sobre sua formação estão mostrados na Tabela 3.12.

Tabela 3.11: *Formação dos 8 bytes de controle.*

Byte	*Bit*	Nome do campo	Descrição
0	7-6	TMCC *identifier*	Identifica que é sistema de TV Digital terrestre (10).
	5	Reservado	Transporta o valor 1 .
	4	*Buffer reset Flag*	Carrega normalmente o valor 0.
	3	Controle de emergência	Durante sinalização de emergência transporta o valor 1 e 0 caso contrário.
	2	Designação de mudança	Transporta normalmente o valor 1. Durante uma troca de parâmetros transporta o valor 0.
	1	Primeiro pacote do quadro	Carrega o valor 1 no primeiro pacote de cada início de quadro de multiplexação.
	0	Indicação do *Frame* OFDM	Durante o quadro OFDM par transporta o valor 0 e 1 caso contrário.
1	7-4	Camada de transmissão do pacote	Indica em qual camada hierárquica do transmissor o pacote será transmitido: 0000 - camada hierárquica inválida 0001 - Camada A 0010 - Camada B 0011 - camada C 1000 - identifica que o pacote transporta a IIP. Este pacote não é transmitido em nenhuma camada hierárquica.
	3-0	Índice de contagem regressiva	Contador que decremeneta seu valor durante o processo de troca de parâmetros de transmissão.
2	7	Sinalizador de parâmetros AC	Caso os dados das portadoras AC estejam sendo transportados nos próximos *bytes*, este sinalizador transporta o valor 0, caso contrário deve transportar o valor 1.
	6-5	Total de *bytes* com dados das portadoras AC	00= 1 *byte*; 01=2 *bytes*; 10=3 *bytes*; 11=4 *bytes* ou quando não há dados AC.
	4-0	Número do pacote no quadro (TSP *Counter*)	Valor de contador que é incrementado a cada pacote do quadro de multiplexação. Inicia seu valor por 0.
3	7-0		
4	7-0	Dados da portadoras AC	Dados da portadoras AC.
5	7-0	Dados da portadoras AC	Dados da portadoras AC.
6	7-0	Dados da portadoras AC	Dados da portadoras AC.
7	7-0	Dados da portadoras AC	Dados da portadoras AC.

- *IIP Branch Number*: caso a informação de rede SFN (*Single Frequency Network*) que é transportada no campo *Network Synchronization Information* ultrapasse 159 *bytes*, mais de um pacote de IIP deverá ser utilizado

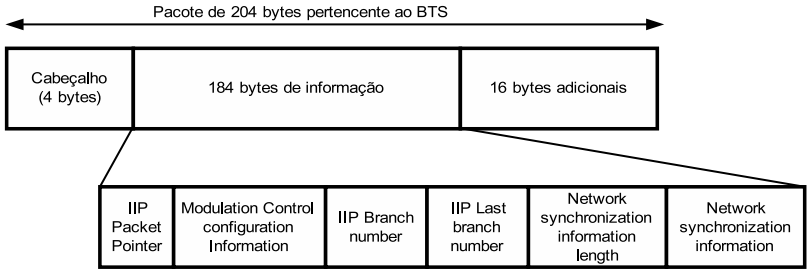


Figura 3.15: *Estrutura de formação do IIP.*

para transportar tal informação. Cada pacote da IIP que transportar parte das informações de rede é chamado de sub-pacote (*branch*). O campo IIP *Branch Number* indica qual o número do sub-pacote em cada IIP, sendo que o primeiro sub-pacote tem o IIP *Branch Number* igual à 0.

- IIP *Last Branch Number*: indica o número do último sub-pacote que transporta as informações de SFN. Caso só exista um sub-pacote, o campo IIP *Last Branch Number* deve transportar o valor 0.

- *Network Synchronization Information Length*: indica a quantidade de *bytes*, logo na sequência, que transportam informações de rede SFN.

- *Network Synchronization Information*: transporta informações de rede SFN. Caso não haja informações sobre rede ou estas não preencham todo o pacote da IIP, deve-se utilizar $0xFF_h$ como enchimento.

As informações transportadas pelo *Modulation Control Configuration Information* estão diretamente relacionadas com os dados transmitidos pelas portadoras TMCC do transmissor. Mais informações sobre as portadoras TMCC podem ser encontradas na Sessão 2.2.12. Na Tabela 3.12 estão mostrados os campos que formam o *Modulation Control Configuration Information*. A Tabela 3.13 mostra a formação do TMCC *Information*, enquanto na Tabela 3.14 pode-se visualizar os parâmetros de transmissão.

3.4 Conclusão

Este capítulo mostra que a etapa de multiplexação de dados envolvem conceitos e conhecimentos profundos a respeito da formação do TS e do BTS. Tabelas e referências de tempo são informações essenciais para a correta decodificação e exibição de um programa e tais informações devem ser alteradas e atualizadas

durante o processo de multiplexação. A estrutura de organização dos dados no BTS são padronizadas e imprescindíveis para sua correta interpretação e transmissão pelo transmissor. As informações apresentadas de forma sintetizada e organizada facilitam o estudo e compreensão dos diversos conceitos comentados e formam uma base para a compreensão da implementação do multiplexador.

Tabela 3.12: *Formação do Modulation Control configuration Information.*

Nome do camapo	Num. *bits*	Descrição
TMCC *Synchronization Word*	1	É utilizado para sincronização do quadro OFDM, já que tem seu valor invertido a cada quadro: no quadro par trasporta 0 e no quadro ímpar transporta 1. Mais detalhes sobre sincronização do quadro OFDM podem ser encontrados na Seção 2.2.12.
AC *Data Effective Position*	1	Indica onde estão sendo transmitidos os dados das portadoras AC: 0 = em pacote enviados em camada hierárquica inválida. 1 = dados transmitidos nos 16 *bytes* adicionais dos pacotes do BTS.
Reserved	2	Reservado para uso futuro.
Initialization Timing Indicator	4	Contador decrescente que indica mudança de modo de transmissão ou intervalo de guarda.
Current Mode	2	Indica qual o modo de transmissão está sendo utilizado no momento. 00: Reservado; 01: modo 1 10: modo 2; 11: modo 3
Current Guard Interval	2	Indica qual o intervalo de guarda que está sendo utilizado no momento. 00: 1/32; 01: 1/16; 10: 1/8; 11: 1/4;
Next Mode	2	Indica o próximo valor para o modo de transmissão.
Next Guard Interval	2	Indica o próximo valor para o intervalo de guarda.
TMCC *Information*	87	Parâmetros da TMCC. Serão vistos com mais detalhes na Tabela 3.13.
Phase Correction of CP	3	Reservado para uso em transmissão de rádio digital.
Reservado para TMCC	10	Reservado para uso futuro da TMCC Corresponde ao *bit* 110 ao 121 da TMCC. Mais detalhes na Tabela 2.10.
Reservado	12	Reservado para uso futuro.
CRC	32	*Bits* de paridade adicionados para possibilitar a detecção de erros nos dados. recebidos.

Tabela 3.13: *Formação do TMCC Information.*

Nome do campo	Num. *bits*	Descrição
System Identifier	2	Idêntico ao especificado na TMCC. Veja a Tabela 2.12.
Count Down	4	Contador decrescente que indica mudança de parâmetros de transmissão. Veja a Tabela 2.13
switch-on control flag for emergency alarm	1	Sinalizador idêntico ao sinalizador de emergência transportado pela TMCC. Veja a Tabela 2.14.
Partial reception flag	1	Indica a transmissão de dados para receptores móveis. Veja a Tabela 2.15.
Transmission Parameters layer A	15	Parâmetros de transmissão da camada A. Veja a Tabela 3.14.
Transmission Parameters layer B	15	Parâmetros de transmissão da camada B. Veja a Tabela 3.14.
Transmission Parameters layer C	15	Parâmetros de transmissão da camada C. Veja a Tabela 3.14.
Next partial reception flag	1	Indica a transmissão de dados para receptores móveis. Veja a Tabela 2.15.
Next transmission Parameters layer A	15	Próximos parâmetros de transmissão da camada A. Veja a Tabela 3.14.
Next transmission Parameters layer B	15	Próximos parâmetros de transmissão da camada B. Veja a Tabela 3.14.
Next transmission Parameters layer C	15	Próximos parâmetros de transmissão da camada C. Veja a Tabela 3.14.

Tabela 3.14: *Formação do Transmission Parameters.*

Nome do campo	Num. *Bits*	Descrição
Esquema de Modulação	3	Indica o esquema de modulação utilizada pela camada hierárquica. Veja a Tabela 2.16.
Taxa do código Convolucional	3	Indica a taxa de código utilizada pela camada hierárquica. Veja a Tabela 2.17.
Profundidade do entrelaçador temporal	3	Especifica a profundidade do entrelaçador temporal utilizado pela camada hierárquica. Veja a Tabela 2.18.
Número de Segmentos	4	Indica o número de segmentos associados a camada hierárquica Veja a Tabela 2.19.

Capítulo 4

Implementação e validação do multiplexador

Neste capítulo estão apresentados detalhes a respeito da implementação de um protótipo do multiplexador para o sistema ISDB-T_B. Na Sessão 4.1 estão mostrados os desafios da implementação. A Sessão 4.2 mostra o modelo adotado, enquanto a Sessão 4.3 mostra as tecnologias e ferramentas utilizadas. A Sessão 4.4 detalha a implementação de um protótipo e a Sessão 4.5 mostra o processo de validação do mesmo. A Sessão 4.6 encerra o capítulo com uma conclusão.

4.1 Desafios da Implementação

Conforme mostrado no capítulo 2, existem diversos pontos de configuração no sistema de transmissão e estes são controlados pelo multiplexador. Portanto, há a necessidade de conhecer o funcionamento do transmissor para a implementação de um multiplexador bem como conhecer o processo de multiplexação de TS, que foi descrito no capítulo 3.

Um dos grande desafios do processo de desenvolvimento do multiplexador foi o entendimento do funcionamento tanto da etapa de transmissão quanto da formação do TS e BTS. Entendimento esse alcançado através do estudo teórico aliado a testes práticos realizados em laboratório. Um segundo desafio, além do estudo e comprovação dessas informações teóricas foi o desenvolvimento de um equipamento que respondesse as necessidades do mercado brasileiro de TV digital. Foi necessário criar um modelo para atender as necessidades estabelecidas pelas normas e escolher tecnologias que atendessem ao modelo adotado. E, após escolher tais tecnologias, estudá-las a ponto de as utilizar para o desenvolvimento de um sistema complexo como é o multiplexador. Um outro desafio estabelecido

era o de oferecer funcionalidades ou facilidades que pudessem fazer com que o multiplexador desenvolvido oferecesse diferenciais perante os multiplexadores importados e que também fosse competitivo com relação ao preço. A forma como foram trabalhados estes desafios, quais as opções e as decisões que foram tomadas estão descritas ao longo deste capítulo.

4.2 Modelo para implementação do sistema de multiplexação

A necessidade do multiplexador realizar diversas tarefas simultâneas que exigem processamento dedicado e precisão com relação as taxas e medidas de tempo foram fatores decisivos na opção pelo FPGA (*Field Programmable Gate Array*) como plataforma de desenvolvimento. Por exemplo, a multiplexação de TS envolve a recepção e processamento paralelo de diferentes TS, além da execução de processo de correção de PCR, que requer precisão de relógio e de medidas de tempo. O FPGA foi escolhido pelo fato desta tecnologia oferecer uma plataforma de processamento que é um *hardware* dedicado ao projeto desenvolvido, diferentemente, por exemplo, do desenvolvimento de uma solução em *software*, que teria de dividir a capacidade de processamento da CPU (*Central Processing Unit*) entre as requisições do sistema operacional, dos periféricos e do próprio *software* do projeto.

A principal linguagem de programação utilizada para implementação do projeto foi a linguagem VHDL (*Very High Speed Integrated Circuit Hardware Description Language*) [20][21] associada a códigos escritos em linguagem C. As ferramentas de desenvolvimento utilizadas foram o *Quartus* II [22] e SOPC (*System-On-a-Programmable-Chip*) *Builder* [23], ambos da empresa Altera [24].

Com relação a configuração do multiplexador, partiu-se da idéia inicial de possibilitar que o mesmo pudesse ser configurado via rede. Buscando a praticidade de não ter de instalar nenhum *software* no computador para a comunicação com o equipamento, optou-se por criar uma interface embarcada. Nesse caso, o usuário precisa somente digitar o endereço IP (*Internet Protocol*) do equipamento em um navegador WEB a partir de um computador qualquer conectado na mesma rede do equipamento, para conseguir acessá-lo e configurá-lo. O fato de não se ter de instalar um *software* pode evitar problemas com relação a compatibilidade do *software* com o computador, como por exemplo, conflito com outros programas instalados, incompatibilidade com diferentes *hardwares* e/ou versões de sistema operacional.

O modelo proposto para implementação do multiplexador pode ser visualizado

na Figura 4.1 [25].

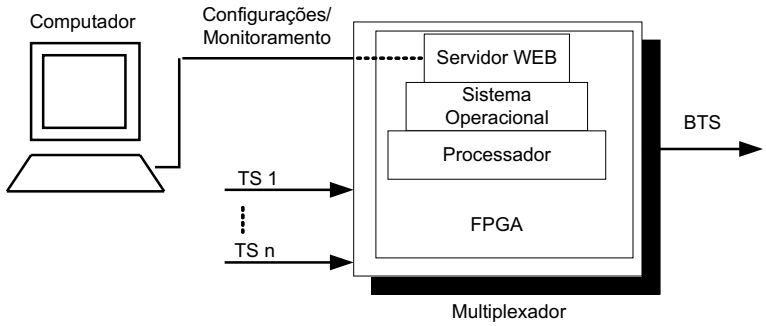


Figura 4.1: *Diagrama do modelo do multiplexador implementado.*

No modelo proposto, a interface com o usuário é construída utilizando linguagem HTML, que precisa ser suportada por um servidor WEB. O servidor WEB, por sua vez, funciona sobre um sistema operacional residente em uma CPU. No modelo implementado, optou-se por utilizar uma CPU embarcada dentro do FPGA evitando-se, dessa forma, a utilização de um processador externo que implicaria em aumento de custo do equipamento.

O servidor WEB utilizado foi o BOA (*Basic Object Adapter*) [26] e o sistema operacional escolhido foi o uClinux [27]. Ambos foram escolhidos principalmente por serem gratuitos e terem sido desenvolvidos para o uso em sistemas embarcados (*embedded systems*).

O processador utilizado é o Nios II [28]. Esse é um processador de propósito geral que foi desenvolvido pela Altera para ser embarcado dentro de seus FPGA's.

O multiplexador deve inserir no seu fluxo de saída novas tabelas compatíveis com o TS MPEG-2 e com sistema ISDB-T_B. Os principais multiplexadores encontrados no mercado desenvolvidos em *hardware* necessitam de um *software* externo para geração dessas tabelas. Nesse *software*, o usuário normalmente deve cadastrar manualmente todas as informações necessárias para o correto funcionamento do sistema.

Na implementação em questão, a geração das principais tabelas do sistema, como a PAT, PMT, NIT e TOT é realizada de maneira automática pelo multiplexador. Ele se encarrega de analisar os fluxos de entrada, extrair as informações de configuração do sistema de transmissão, a partir da interface WEB, e gerar as tabelas necessárias. Uma grande vantagem dessa abordagem é a facilidade de operação do equipamento e o fato de se ter uma grande versatilidade nas alterações do sistema, como reconfigurar algum codificador de vídeo ou alterar alguma informação do sistema de transmissão. Assim que detectada qualquer alteração em qualquer desses parâmetros, novas tabelas são geradas levando em consideração sempre as últimas informações.

A geração dessas tabelas é uma tarefa complexa de ser implementada utilizando a linguagem VHDL devido a necessidade de se trabalhar com manipulação de vários vetores de informação em paralelo para gerar um vetor final de saída. Optou-se, então, pelo desenvolvimento da geração das tabelas utilizando a linguagem C. Esses códigos são executados pela CPU embarcada no FPGA.

4.3 Técnicas e Ferramentas para Implementação do multiplexador

A seguir, apresenta-se uma breve descrição das tecnologias, linguagens e *softwares* utilizados no processo de desenvolvimento do protótipo do multiplexador.

4.3.1 *Hardware* utilizado

A Figura 4.2 apresenta uma foto do *hardware* utilizado para a implementação do protótipo do multiplexador. Foi utilizado como base um *kit* de desenvolvimento da Altera, o Nios II *Development Kit* - EP2S60 [29], associado a uma placa de 4 entradas e 2 saídas ASI (*Asynchronous Serial Interface*) [30] desenvolvida pela empresa Linear Equipamentos Eletrônicos S.A [31]. A escolha pelos FPGAs da Altera se deve ao fato desse dispositivo já ser empregado por essa empresa em outros equipamentos. Como o custo do FPGA diminui de acordo com a quantidade de peças compradas, optou-se pela utilização de FPGAs semelhantes aos já utilizados dentro da empresa, visando menor custo final do equipamento. A implementação a partir de um *kit* de desenvolvimento foi utilizada para que, pelo menos num primeiro momento, não houvesse preocupação com o desenvolvimento de placas de circuito impresso e com problemas relacionados ao desenvolvimento de um *hardware*.

O FPGA empregado nesse *kit* é o modelo EP2S60F672 [32] pertencente a família Stratix II da Altera. Esse componente possui 60.440 elementos lógicos e 2.544.192 *bits* de memória RAM (*Random Access Memory*).

4.3.2 Tecnologia FPGA

O primeiro FPGA comercializado foi desenvolvido pela empresa Xilinx Inc. no ano de 1983. Esse consiste de um dispositivo lógico programável (*Programmable Logic Device* - PLD) que suporta a implementação de circuitos lógicos relativamente grandes [33].

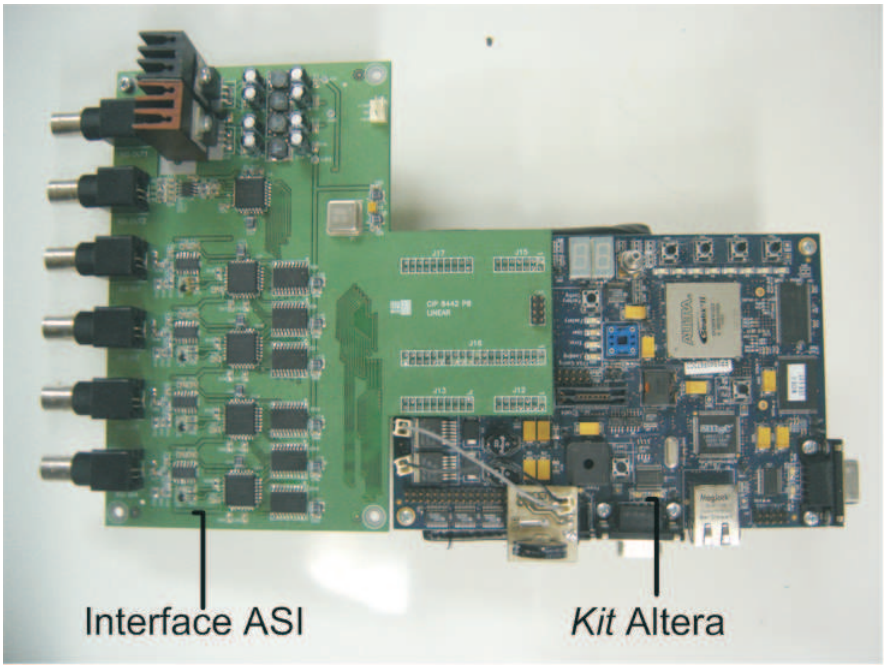


Figura 4.2: *Foto do hardware utilizado na implementação.*

O FPGA se destaca na implementação de circuitos digitais, oferecendo uma boa relação custo/benefício. Ele oferece maior flexibilidade que microprocessadores de uso geral e menor custo que os circuitos integrados de aplicação específica (*Application Specific Integrated Circuit* - ASIC) [34]. Normalmente as aplicações implementadas em FPGA'S são mais lentas e consomem mais energia que as aplicações implementadas em um ASIC. Porém, o custo envolvido na fabricação de um ASIC só é justificável em caso de produção em larga escala. O tempo envolvido no desenvolvimento de um ASIC também é maior que o tempo gasto em uma aplicação desenvolvida para um FPGA [35].

O FPGA é composto por uma matriz de blocos ou elementos lógicos. Cada um desses elementos lógicos é capaz de realizar pequenas funções e operações lógicas que podem ser programados ou configurados de acordo com a função lógica que irão empenhar. Os elementos lógicos são interconectados entre si de maneira a se obter o resultado lógico final desejado em cada projeto implementado.

Pode-se visualizar na Figura 4.3 a estrutura básica de formação de um FPGA, que é composto por três tipos principais de recursos: os blocos lógicos, os blocos de entrada e saída (I/O - *Input/Output*), e as chaves programáveis de interconexão [33],[34],[35].

Os blocos lógicos são organizados em linhas e colunas e as chaves de interconexão programáveis são dispostas de forma a possibilitar a conexão dos blocos lógicos da maneira que for conveniente, além de conectar estes aos blocos de I/O . Quando se implementa um circuito em FPGA, os blocos lógicos são configurados

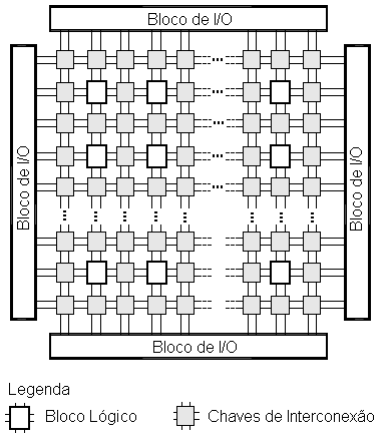


Figura 4.3: *Estrutura básica de formação de um FPGA.*

de forma a realizar as operações lógicas necessárias e as chaves de interconexão estabelecem as conexões entre os blocos de maneira a se obter o resultado lógico final requerido pelo projeto.

4.3.3 Etapas de desenvolvimento de projetos em FPGA

O objetivo final, quando se cria um projeto em FPGA, é o de programar os elementos digitais que o formam de maneira que seja criado um circuito capaz de executar uma determinada funcionalidade. A elaboração de projetos utilizando FPGA's possuem três principais etapas, que são a etapa de descrição, a etapa síntese e a etapa de posicionamento e interligação. Essas etapas podem ser realizadas por programas diferentes e separados ou por um único programa que integre todas as etapas, como é o caso do *Quartus* II da Altera, que foi utilizado como ferramenta de desenvolvimento do multiplexador.

Antes de mais nada, é necessário realizar a especificação do projeto, listando as funcionalidades e tarefas que o projeto tem por objetivo realizar. Na etapa de descrição, as especificações do projeto são escritas em uma linguagem de descrição de *hardware* como VHDL ou *Verilog* [36], resultando em um código que pode ser compilado e simulado. O código compilado é interpretado por uma ferramenta de síntese que irá transcrever o código em um circuito equivalente que utilize elementos básicos e genéricos como registradores, portas lógicas, somadores e comparadores. O circuito gerado nesta primeira fase da síntese está em um nível chamado de RTL (*Register Transfer Level*). Em seguida, a ferramenta de síntese reescreve o circuito em nível RTL utilizando elementos específicos do dispositivo em que o projeto será implementado, substituindo os elementos básicos. Um arquivo contendo o esquema de ligação entre os elementos do dispositivo é ger-

ado. Este esquema se limita a listar os dispositivos usados e quais as conexões entre eles. Na sequência, já na etapa de posicionamento e interligação, é defindo quais serão os dispositivos dentro do FPGA que serão utilizados. Define-se, por exemplo, a localização ou posicionamento de cada um deles dentro do FPGA (*place*) e como será a interligação ou roteamento (*route*) das ligações entre esses dispositivos [20], [33], [34]. Pode-se gerar, após essa fase, o arquivo binário a ser gravado no FPGA. Simulações podem ser realizadas antes da gravação do projeto no FPGA com o objetivo de se verificar se as funcionalidades desejadas foram corretamente descritas e implementadas.

4.3.4 VHDL

As duas principais linguagens de descrição de *hardware* mais utilizadas são o VHDL e o *Verilog*. A linguagem VHDL necessita de mais linhas de código para descrever uma funcionalidade do que um código *Verilog* equivalente. Essa é uma característica intencional da linguagem VHDL, que foi criada com intuito de servir também como parte da documentação de um projeto de *hardware*. E por fim, a linguagem possui uma semântica para simulação menos ambígua. A ambiguidade do *Verilog* pode fazer com que resultados diferentes sejam encontrados ao se utilizar diferentes ferramentas de simulação, o que não deve acontecer em uma simulação de um código VHDL [37]. Por essas razões, o VHDL foi escolhido como linguagem de implementação do multiplexador.

A Linguagem VHDL foi desenvolvida pelo Departamento de Defesa dos Estados Unidos da América (DARPA). Eles buscavam desenvolver uma linguagem de descrição de *hardware* que forçasse o desenvolvedor a escrever um código mais estruturado e compreensível para que, dessa forma, o próprio código servisse como um documento de especificação do projeto [20],[38]. A linguagem VHDL foi padronizada pelo IEEE (*Institute of Electrical and Electronic Engineers*) em 1987 através da criação do padrão IEEE 1076-1987. Em 1993 foi publicada uma versão revisada desse padrão, que deu origem ao padrão IEEE 1076-1993, que é o mais utilizado atualmente [20].

Uma linguagem de descrição de *hardware* é uma forma de se descrever ou especificar, através de um programa, o comportamento de um circuito. Ela deve descrever o que o circuito faz e de que maneira ele o faz. O *hardware* pode se referir tanto a sistemas complexos, como um microprocessador, quanto a sistemas simples, como uma porta lógica. A linguagem VHDL abrange diversas escalas de aplicações podendo ser utilizada para descrever o funcionamento de *hardware* em geral [38].

4.3.5 O ambiente de desenvolvimento

O *software* utilizado para implementar e simular as etapas desenvolvidas em VHDL foi o *Quartus* II, desenvolvido pela Altera. Ele foi escolhido por ser uma ferramenta que integra todas as etapas de desenvolvimento de projetos em FPGA (descrição, síntese e posicionamento e interligação) [20], além de oferecer um ambiente de simulação e de já ter integrado ferramentas para criação de processadores embarcados. Outro fator positivo é que o *Quartus* foi desenvolvido pela própria Altera, que é o fabricante do FPGA utilizado. Pode-se, através do ambiente de simulações, estabelecer sinais na entrada do projeto e analisar os resultados obtidos na saída do mesmo. O *Quartus* gera um arquivo final de gravação do projeto no FPGA e oferece a ferramenta para a gravação do mesmo. Na Figura 4.4 tem-se uma visão da interface do *software*. Nessa, podemos visualizar a representação em esquemático do projeto. Nesse ambiente, é possível efetuar a interconexão de diferentes módulos ou etapas entre si, a módulos prontos (megafunções) ou outros recursos oferecidos pelo *software*, como portas lógicas, contadores, etc. Também é possível visualizar na figura também a janela onde é elaborada a programação de descrição de *hardware* do projeto.

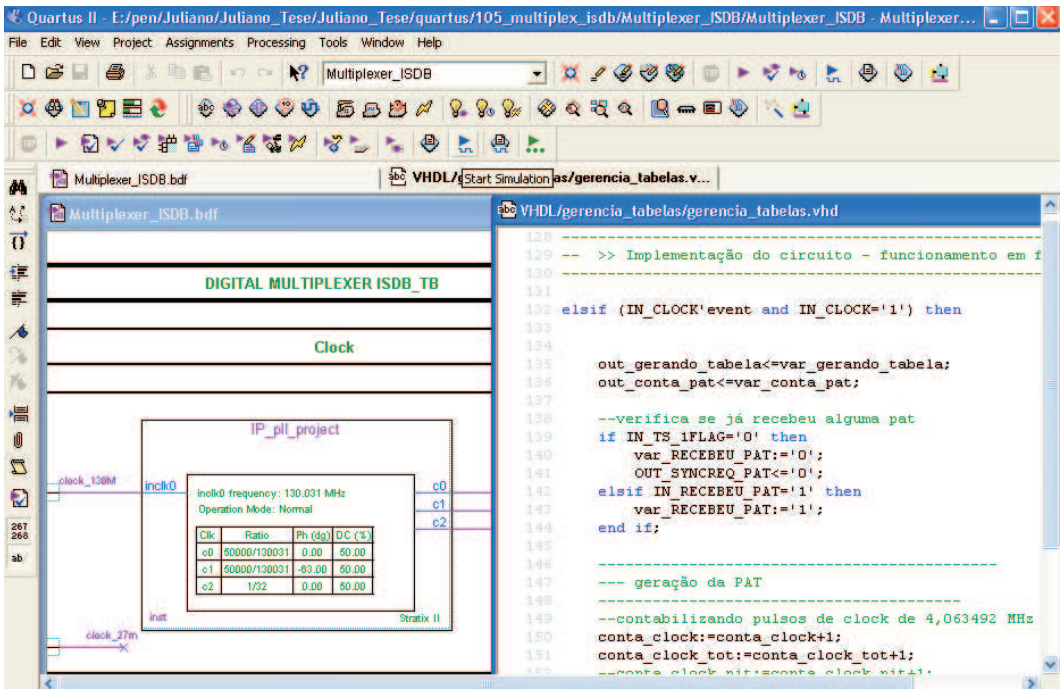


Figura 4.4: *Visão do software Quartus II da Altera.*

O *Quartus* oferece uma ferramenta chamada de *SignalTap*, que é apresentada na Figura 4.5. Através dessa ferramenta é possível capturar os sinais dentro do FPGA, o que permite monitorar e verificar o funcionamento de um projeto implementado. Essa ferramenta foi utilizada durante o processo de desenvolvimento e

será usada para demonstração de resultados e funcionamento da implementação do multiplexador. Para a utilização do *SignalTap* é necessário especificar quais são os sinais que se deseja visualizar, qual é o sinal de referência (*clock*) utilizado para a amostragem e a quantidade de amostras que serão tomadas. É possível especificar condições específicas de amostragem dos sinais através de um gatilho (*trigger*). É necessário que haja memória interna do FPGA disponível para que as amostras dos sinais possam ser armazenadas antes de seu envio para o *software*. A comunicação do *software* com o FPGA é feita através de uma interface JTAG (*Joint Test Action Group*) [39] conectada via porta USB ao PC. A JTAG é uma interface padronizada para testes e programação de dispositivos e é utilizada também para a gravação do FPGA.

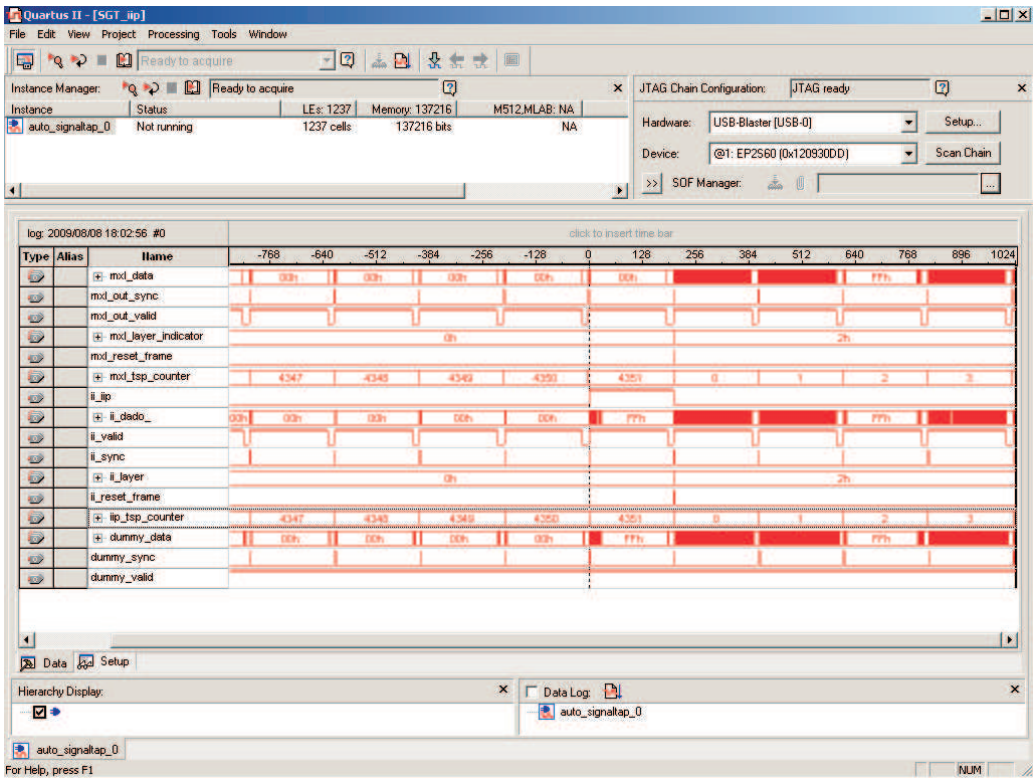


Figura 4.5: *Visão da ferramenta SignalTap.*

A transformação via *software* da descrição menos detalhada (código desenvolvido) numa informação mais precisa e elaborada é a chamada de síntese. Algumas ferramentas de síntese são capazes de mapear a linguagem de descrição de *hardware* diretamente em determinados circuitos integrados [38], como é o caso da ferramenta *Quartus* II, que possibilita a síntese do projeto direcionada à tecnologia FPGA.

Para a criação do processador embarcado foi utilizado o *software* SPOC *Builder*, que é uma ferramenta do *Quartus* II. A finalidade dessa ferramenta é a de possibilitar a geração de sistemas integrados em um único dispositivo ou SOPC [23].

Utilizando o SOPC *Builder* pode-se especificar o processador Nios II, bem como seus periféricos a serem implementados dentro do FPGA.

Para o desenvolvimento de aplicações a serem executadas diretamente pelo processador, foi utilizada outra ferramenta do *Quartus* II, o Nios II IDE (*Integrated Development Environment*), mostrada na Figura 4.6. O Nios II IDE é o sistema integrado para o desenvolvimento de *software* para o Nios II, que inclui o IDE *Eclipse* e compilador C/C++.

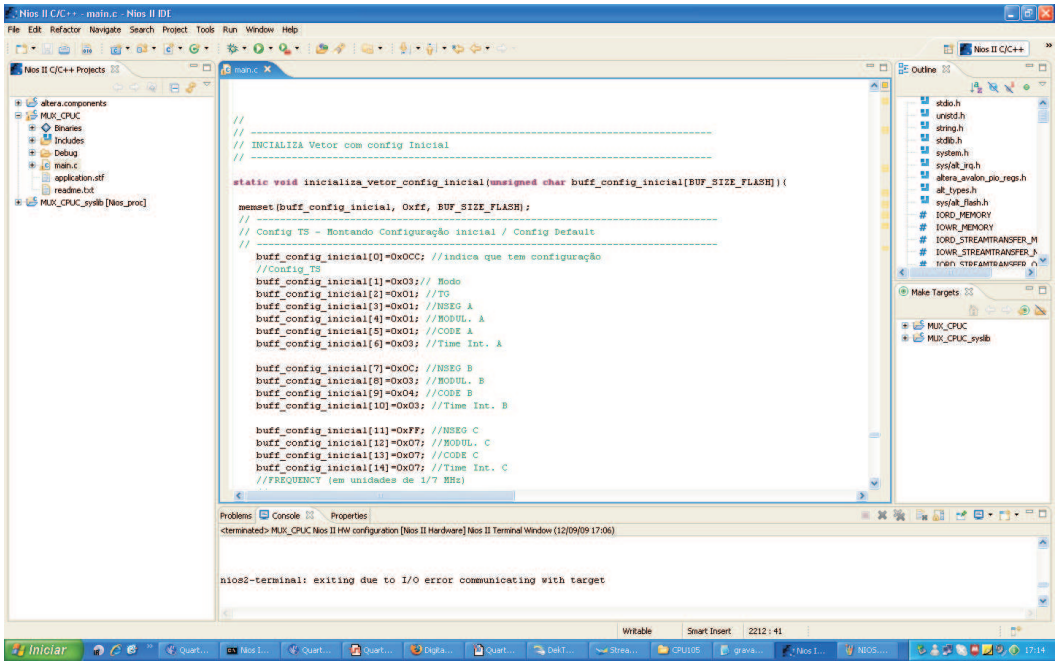


Figura 4.6: *Visão da ferramenta Nios II IDE.*

4.3.6 Processador Nios II

O processador Nios II é um processador RISC (*Reduced Instruction Set Computer*) implementado usando linguagem de descrição de *hardware* para ser sintetizado dentro do FPGA. Ele é classificado como sendo um *soft-core processor* já que o processador não está fixo no silício, ou seja, não foi implementado durante a confecção da pastilha de silício do FPGA. Ele é configurável, uma vez que pode-se selecionar as características do processador bem como seus periféricos. Através da ferramenta SOPC *Builder* pode-se adicionar diversos periféricos padrões, que já estão prontos, como interfaces de comunicação serial, interfaces com memórias, relógios (*timers*) e outros periféricos de propósito geral. Além disso, há a possibilidade de se criar periféricos personalizados e instruções também personalizadas (é possível fazer uso de até 256 instruções personalizadas) [28]. Essas facilidades oferecidas permitem uma grande versatilidade e flexibilidade durante o desen-

volvimento de projetos. Na Tabela 4.1 estão listadas as principais características do processador Nios II [28].

Tabela 4.1: *Principais características do processador Nios II.*

Características
Conjunto de instruções de 32 *bits*.
Barramento de dados e espaço de endereçamento de 32 *bits*.
32 registradores de propósito geral.
5 registradores de controle.
32 fontes de interrupção externa.
Possui Unidade de Gerenciamento de Memória (MMU - *Memory Management Unit*) opcional.
Desempenho pode chegar a 250 DMIPS (*Dhrystone million instructions per second*).

O *software* SOPC *Builder* possibilita escolher 3 possíveis configurações padrões para a CPU do Nios II, como pode ser visualizado na Figura 4.7. As 3 opções oferecidas são: econômica (*economic*), padrão (*standard*) e rápida (*fast*). A CPU Nios II/e é a que utiliza a menor quantidade de recursos (*economic*) do FPGA, mas em contrapartida é a que possui menor desempenho. Já a CPU Nios II/f apresenta o melhor desempenho (*fast*) em detrimento do uso de maior número de recurso do FPGA. A CPU Nios II/s (*standard*) oferece um meio termo em relação ao desempenho e uso de recursos.

A opção de se utilizar a CPU embarcada dentro do FPGA faz com que não seja necessário o investimento na compra de um processador dedicado, e diminui, consequentemente, o número de componentes na placa de circuito impresso, diminuindo também a sua complexidade e preço do multiplexador. Mas a utilização da CPU embarcada no FPGA, em contrapartida, resulta em uma maior utilização de lógica do FPGA, o que pode resultar no uso de um dispositivo mais denso e, portanto, mais caro.

4.3.7 Sistema operacional *uClinux*

A utilização de um sistema operacional compacto e livre de custos foram características que levaram a escolha pelo sistema operacional *uClinux* [40].

O sistema operacional *uClinux* é uma variante do *Linux* que foi desenvolvida para ser utilizada em sistemas embarcados. Mais especificamente, foi desenvolvido visando o uso em microcontroladores de baixo custo. Um fator que contribui para o baixo custo dos microcontroladores é o fato de, comumente, não se oferecer suporte à MMU. O *uClinux* é, por esse motivo, um sistema operacional que não

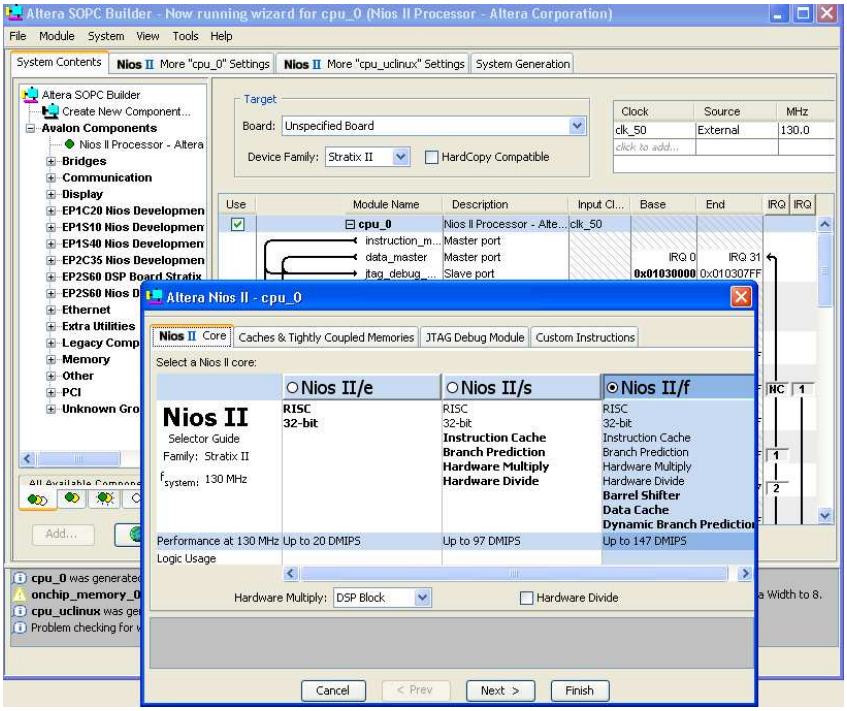


Figura 4.7: *Configuração da CPU do Nios no software SOPC Builder.*

faz uso de MMU [40],[41]. Enquanto o *Linux*, fazendo uso da MMU, provê um endereçamento virtual de memória para cada processo a ser executado, o *uClinux* oferece um único espaço de endereço que é compartilhado entre todos os processos. Isso faz com que o gerenciamento de memória no *uClinux* fique mais complexo, mas possibilita que ele seja mais compacto [41].

O projeto do *uClinux* teve início em 1997 e, em 1998, teve sua primeira versão disponibilizada publicamente, sendo proposta como um sistema operacional alternativo para o *Palm Pilot*. Atualmente o *uClinux* vem ganhando cada vez mais espaço dentre as aplicações embarcadas, sendo visto como um dos mais populares sistemas operacionais para tal aplicação. Uma característica que contribui para este sucesso é o fato dele ser um sistema gratuito e de código aberto (*open-source*) além do fato de ser suportado em diversas plataformas como ARM (*Advanced RISC Machine*), *ColdFire*, NIOS entre outras. O sistema operacional oferece suporte a vários protocolos de redes, incluindo a pilha TCP/IP (*Transmission Control Protocol / Internet Protocol*) e sistemas de arquivos.

4.3.8 O servidor WEB BOA

O servidor BOA é um servidor WEB de código livre, gratuito e que foi desenvolvido para uso em aplicações embarcadas [26]. Essas características foram de-

cisivas para a escolha desse servidor, aliadas ao fato dele já vir como uma opção de instalação no sistema *uClinux*, o que facilita sua utilização.

4.4 Implementação do sistema de multiplexação

Nessa sessão, mostra-se com mais detalhes a implementação do multiplexador. Um diagrama resumido do projeto implementado pode ser visualizado na Figura 4.8.

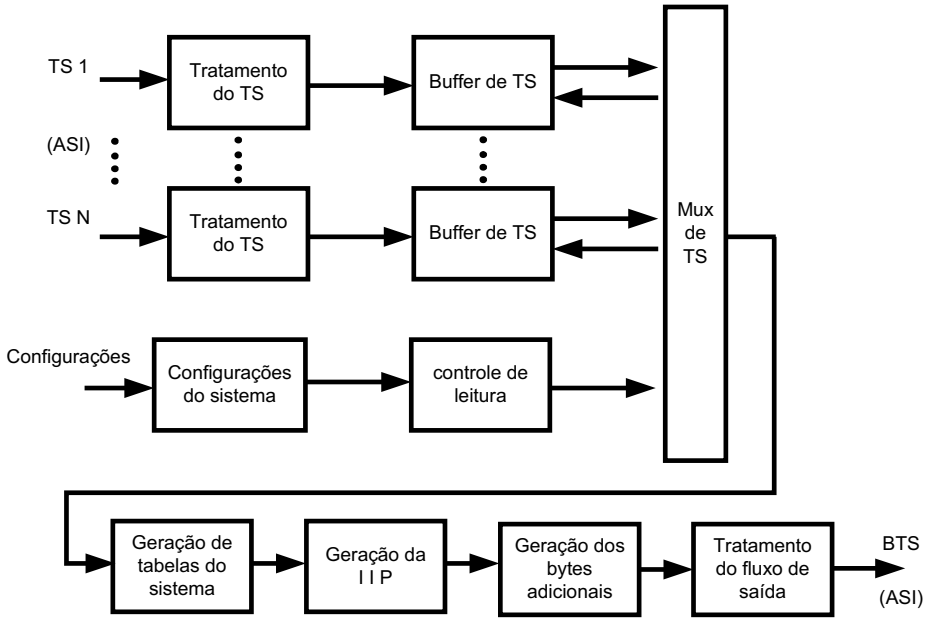


Figura 4.8: *Diagrama do multiplexador implementado.*

Cada uma das etapas que aparecem neste diagrama será abordado com mais detalhes a seguir.

4.4.1 Tratamento do TS de entrada

Essa etapa é responsável por receber o TS através da interface ASI e retirar as informações necessárias para o sistema gerar novas tabelas de transmissão. Essa etapa também realiza parte do tratamento de PCR. O sinal entregue por esta etapa é encaminhada ao *buffer* de TS. A Figura 4.9 mostra as etapas que compõe o tratamento do TS de entrada, descritas sucintamente a seguir.

Interface ASI de entrada

A interface ASI é uma interface de comunicação assíncrona, que possui uma taxa fixa de 270 Mbps e codifica os dados transmitidos utilizando um código 8B10B

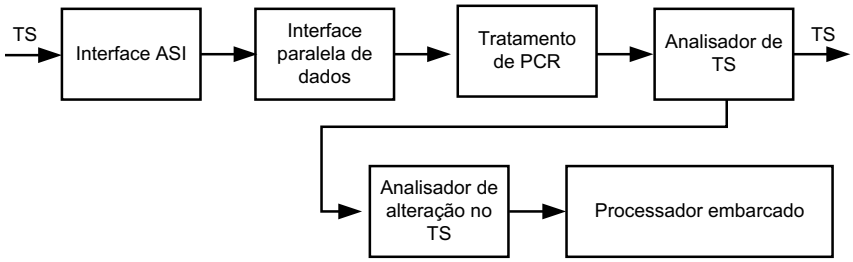


Figura 4.9: *Tratamento do TS de entrada.*

[30]. Uma placa de circuito impresso foi criada especificamente para o tratamento dos sinais recebidos da ASI. Essa placa serve como a entrada e saída de dados do protótipo no formato compatível com o padrão ASI. Para a interface ASI de entrada, ela realiza a decodificação 8B10B e entrega os dados organizados em *bytes* tendo como referência um *clock* de 27 MHz. A interface entrega, juntamente com os dados, uma sinalização de dado válido repassados ao FPGA, conforme mostra a Figura 4.10. A placa também converte o fluxo de dados entregue pelo FPGA em uma interface ASI de saída, conforme está descrito na Sessão 4.4.9.

Interface Paralela de Dados

Para que se possa localizar corretamente os dados dentro dos pacotes do TS é necessário saber onde é o início de cada pacote. Essa etapa é a responsável por detectar o sincronismo dos pacotes do TS e entregar em sua saída os dados em uma interface paralela de dados que é formada por 3 sinais: dados, sinal de habilitação de dados válidos e sinalização do *byte* de sincronismo no início de cada pacote do TS. Essa etapa sinaliza ainda a presença ou ausência de TS em sua entrada. A busca pelo sincronismo do TS é iniciada a partir do momento em que é localizado o primeiro *byte* 47_h [10]. A partir desse ponto, é verificado se no 188º ou no 204º *byte* seguinte possui esse mesmo valor. Se for confirmado, o teste prossegue para o próximo pacote. Caso sejam encontradas 5 confirmações em sequência, admite-se que o sincronismo de início de pacote foi localizado. Testes foram realizados considerando-se apenas 3 confirmações em sequência e o resultado também foi satisfatório. Porém, seguindo como orientação o padrão DVB [16], utilizou-se 5 confirmações. A partir do instante em que se localizou o sincronismo do TS, os dados passam a ser entregues pela Interface Paralela de Dados às etapas seguintes. O teste de sincronismo prossegue durante todo o recebimento dos dados, para verificar se não houve perda de sincronismo do TS recebido. Caso aconteça mais de 3 erros de sincronismo consecutivos, admite-se que houve perda de sincronismo com o TS e, nesse instante, a etapa deixa de repassar dados para sua saída. Testes foram realizados utilizando-se tanto TS's

formados por pacotes de 188 quanto de 204 *bytes*.

Na Figura 4.10 está mostrada a análise no *SignalTap* da saída e da entrada da Interface Paralela de Dados. Foi aplicado na entrada do protótipo do multiplexador um TS utilizando-se uma placa da empresa Dektec [42] que, conectada ao computador, permite reproduzir e analisar um TS via interface ASI. Para execução de tais funcionalidades, são utilizados os *softwares StremXpress* [43] e *StreamXpert* [44] respectivamente. Os dados do TS repassados pela Interface ASI podem ser visualizados nos sinais *asi_data* e *asi_valid*. Na amostra 0 da análise está o instante em que a etapa sinaliza que encontrou corretamente o sincronismo do fluxo de entrada (sinal *ss_sincronizado* passa a valer 1). Nesse instante, a etapa começa a entregar em sua saída os dados organizados em uma interface paralela de dados: *ss_data*, *ss_sync* e *ss_valid*. Os sinais com nomenclatura *valid* são os que indicam dados válidos. Esta sinalização acontece quando esse sinal está em nível 1, com exceção do sinal *valid* recebido da interface ASI, que funciona com lógica invertida.

Tratamento do PCR

O PCR é uma referência de tempo transportada dentro de pacotes do TS. Essa referência possibilita a correta decodificação e exibição do sinal de áudio e vídeo pelo receptor. O valor do PCR está diretamente relacionado com a taxa do vídeo e com a quantidade de *bytes* válidos transmitidos entre pacotes consecutivos com PCR, conforme mostrado na Sessão 4.4.1. Quando ocorrem remoções ou inserções de pacotes em um TS, altera-se o espaçamento original de seus pacotes. É o que ocorre na multiplexação de TS. Nesse caso, há a necessidade de se realizar atualização do valor do PCR, para que esse esteja de acordo com a nova organização e taxa do fluxo multiplexado.

Na implementação do multiplexador, os pacotes nulos recebidos do TS são descartados para evitar que a capacidade máxima de armazenamento dos *buffers* seja atingida devido a pacotes sem informação útil. O multiplexador deve transmitir o TS resultante do processo de multiplexação em uma taxa fixa de aproximadamente 32,508 Mbps. No processo de leitura dos *buffers* para gerar o TS de saída do multiplexador, eventualmente, é necessário gerar pacotes nulos para garantir a taxa exata de transmissão requerida pela configuração do sistema de transmissão. Tanto a retirada quanto a inserção de pacotes nulos faz com que a distância entre pacotes com PCR sejam alteradas. Portanto, há a necessidade de se atualizar o valor do PCR para que esse esteja de acordo com a nova organização e taxa do TS resultante.

A técnica utilizada para atualização do PCR é feita em duas fases. A primeira fase acontece quando o pacote que transporta o PCR entra no sistema e a segunda

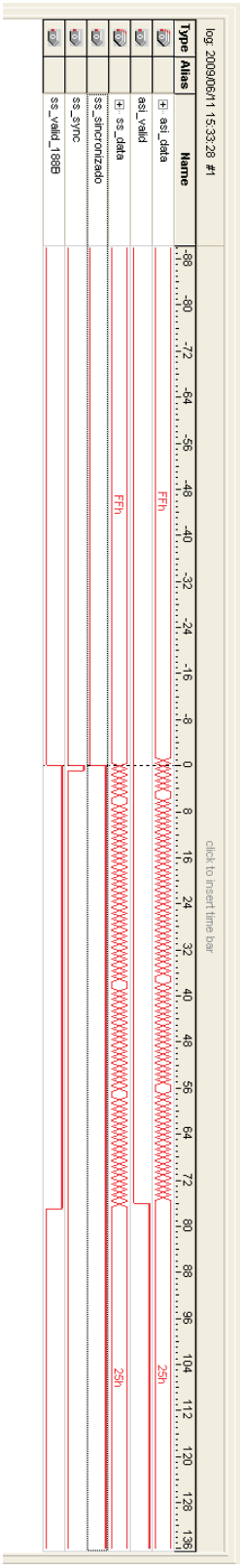


Figura 4.10: *A interface paralela de dados no SignalTap.*

quando o pacote é entregue à saída [45]. Na primeira fase, o valor do PCR do pacote recebido é subtraído do valor de um PCR local que é gerado a partir de um *clock* de 27 MHz localizado na placa do protótipo. Na segunda, o valor do PCR do pacote é somado ao valor do PCR local. Essa técnica acrescenta cerca de 50 ns no valor do *jitter* estático do PCR que, por norma, não deve exceder 500 ns [45]. Essa etapa implementa a primeira fase da atualização de PCR, conforme pode ser visualizado na Figura 4.11.

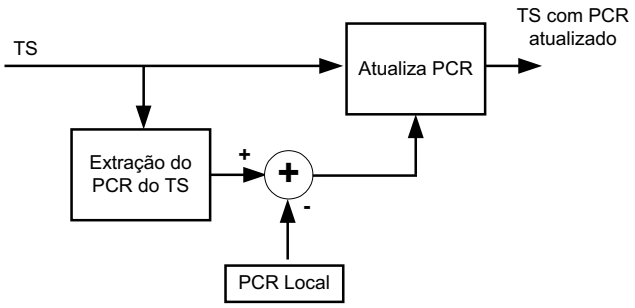


Figura 4.11: *Primeira fase do tratamento do PCR.*

A Figura 4.12 apresenta as amostras de dados capturadas com o *SignalTap* referentes a etapa implementada. Os sinais mostrados cujo nome se inicia por ss (*ss_data*, *ss_sync* e *ss_valid_188B*), são os sinais de entrada da etapa de Tratamento do PCR. Já os sinais de saída desta etapa tem seu nome iniciado por pcr1 (*pcr1_data*, *pcr1_sync* e *pcr1_valid* e *pcr1_pcr_flag*). Na amostra de número -7 pode-se visualizar o primeiro *byte* de um pacote, *ss_sync* assume valor '1' indicando o *byte* de sincronismo, que está sendo entregue à etapa de tratamento de PCR. Do quarto *byte* desse pacote, cujo valor é $2F_h$(ver amostra de número -5 do sinal *ss_data*), pode-se extrair o campo *adaptation field control* que tem valor igual a 10_b, indicando que o pacote possui *adaptation field* mas não possui carga, conforme indicado na Tabela 3.2. Quando esse pacote é entregue na saída da etapa de tratamento de PCR (ver dados após a amostra número 0) é sinalizada a presença de PCR no pacote através do sinal *pcr1_pcr_flag* que assume nível '1'. E quando é detectada a presença de PCR em uma pacote, a etapa realiza o algorítimo referente a primeira fase de tratamento de PCR.

Na Figura 4.12 pode-se visualizar os valores do PCR local, mostrados através dos sinais pcr_local_base, que mostra o valor do PCR *Base*, e o sinal pcr_local_ext, que mostra o valor do PCR *Extension*. Os valores de PCR extraídos do pacote do TS são mostrados nos sinais *pcr_base_or* e *pcr_ext_or*. O valor final do PCR atualizado, após a realização da primeira fase da atualização de PCR, pode ser visto nos sinais *pcr_base_final* e *pcr_ext_final*. O valor final do PCR *Base*, considerando os valores extraídos do pacote, mostrados na amostra de número 5, é dado por

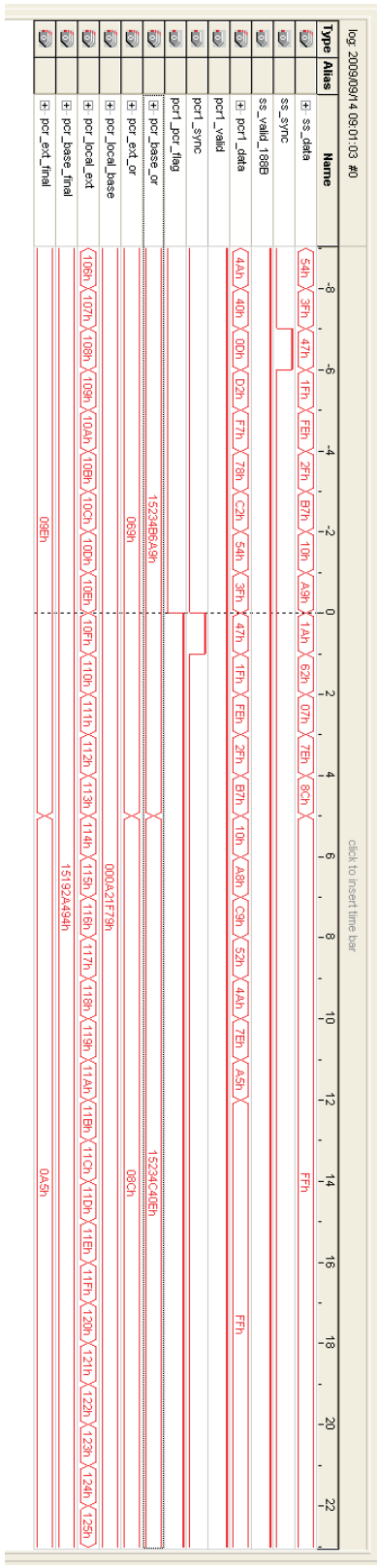


Figura 4.12: *Primeira etapa do tratamento de PCR vista no SignalTap.*

$$\begin{aligned} PCR_{basefinal} &= pcr_base_or - pcr_local_base \\ PCR_{basefinal} &= 15234C40E_h - 000A21F79_h = 15192A495_h, \end{aligned} \tag{4.1}$$

que convertido para decimal fica igual a 5.663.532.181. Esse valor pode ser observado no campo *pcr_base_final* no instante de amostragem igual a 5. Caso o valor seja negativo, isso indica que o valor do PCR original está menor que o valor do PCR Local. Para transformar este valor em um valor positivo, deve-se subtrair o valor encontrado do valor máximo admissível para o PCR *Base* (8589934591_d ou $1FFFFFFFF_h$). O valor calculado diferencia em uma unidade do valor mostrado na Figura 4.12. Isso indica que o valor do PCR *extension* calculado terá um valor negativo.

De forma análoga, para o PCR *extension* tem-se

$$PCR_{extfinal} = pcr_ext_or - pcr_local_ext \tag{4.2}$$

Como o valor do PCR *extension* extraído do pacote ($08C_h$) é menor que o valor do PCR *extension* local (113_h), o resultado será um valor negativo. Calculando o valor do PCR *extension* final, fica

$$PCR_{extfinal} = 140_d - 275_d = -135_d. \tag{4.3}$$

Para transformar esse valor em um valor positivo, deve-se subtrair do valor máximo admissível para o PCR *extension* (300 ou $12C_h$). O valor equivalente ao encontrado é dado por

$$PCR_{extfinal} = 300_d - 135_d = 165_d. \tag{4.4}$$

O resultado obtido,equivalente a $A5_h$, é o mesmo obtido no sinal *pcr_ext_final* da amostra de valor 5 da Figura 4.12. A etapa de tratamento de PCR substitui os valores de PCR transportados pelo pacote pelos novos valores calculados.

Analisador de TS

Essa etapa é responsável por duas tarefas: a de extrair informações importantes do fluxo para serem utilizadas na geração de novas tabelas como a PAT e NIT e pelo descarte de pacotes do fluxo. Estas tarefas são realizadas utilizando-se como principal informação o PID de cada pacote do fluxo. Informações importantes como *program numbers* dos programas do fluxo e PID's das respectivas PMT's são extraídos do conteúdo do pacote que transporta a PAT. Na Figura 4.13 está mostrado o instante em que as informações são repassadas à saída do analisador de TS. No instante de amostragem zero pode-se visualizar a saída de uma tabela

PAT nos sinais *ts_data*, *ts_sync* e *ts_valid*. O sinal *ts_pat_flag* indica a saída de uma tabela PAT e, consequentemente, a saída das informações relativas aos programas existentes no TS no sinal *ts_infots*.

Levando-se em consideração a organização dos dados no sinal *ts_infots* que está mostrada na Figura 4.14, e analisando as informações mostradas em 4.13, pode-se concluir que o fluxo sob análise transporta um TS cujo TS ID é igual a $7FE0_h$ e que a quantidade de programas no TS é igual a 1, isso desconsiderando a NIT que é listada na PAT. O *program number* do programa transportado é 400_h (1024_d), mostrado após a amostra 4 da Figura 4.13, e o PID da PMT associada a esse programa é $E020_h$ (32_d), mostrado após a amostra número 6 da mesma figura.

Na Figura 4.15 é possível visualizar a saída do multiplexador, no *software Streamxpert*, considerando apenas o TS com *program number* 0400h aplicado em sua entrada. A tabela PAT de saída do multiplexador foi criada baseada nas informações recebidas do sinal *ts_infots*. No *software* os PID's e *program numbers* são exibidos em decimal. Cada TS aplicado à entrada do multiplexador irá gerar um sinal *ts_infots* correspondente. Todos esses sinais são enviados à etapa analisador de alterações no TS, que é a responsável por repassar informações à CPU Nios II que irá processá-las para gerar uma única PAT correspondente.

O analisador de TS descarta pacotes de entrada que possam gerar conflitos no fluxo de saída. Por exemplo, os pacotes que transportam a PAT, NIT e TOT devem ser descartados, pois tais informações serão re-geradas pelo multiplexador. Essa etapa realiza também o descarte de pacotes nulos, para que não haja sobrecarga do *buffer* de TS devido ao acúmulo de pacotes não úteis.

Analisador de alterações no TS

As informações extraídas pelo analisador de TS são monitoradas por essa etapa. O monitoramento é realizado de maneira contínua pelo analisador de TS que envia a cada 100 ms em média (intervalo de recepção da PAT) novas informações retiradas do fluxo. O analisador de alterações no TS recebe tais informações e compara com as armazenadas em seu *buffer*. Caso sejam encontradas diferenças, ocorre a atualização no *buffer* e uma sinalização de novos dados recebidos é gerada. Sempre que novas informações são recebidas, essas são repassadas à CPU embarcada, que é a responsável por gerar as novas tabelas de saída do multiplexador baseadas nas últimas informações extraídas dos fluxos de entrada.

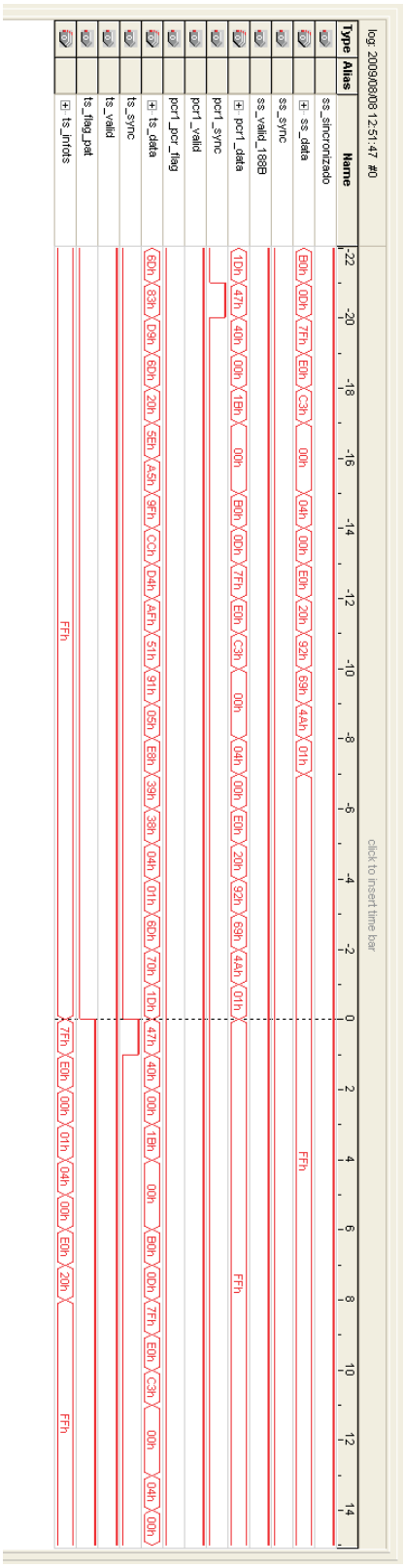


Figura 4.13: *O analisador de TS no SignalTap.*

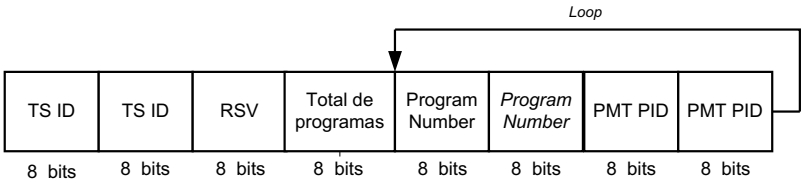


Figura 4.14: *Organização dos dados extraídos da PAT.*

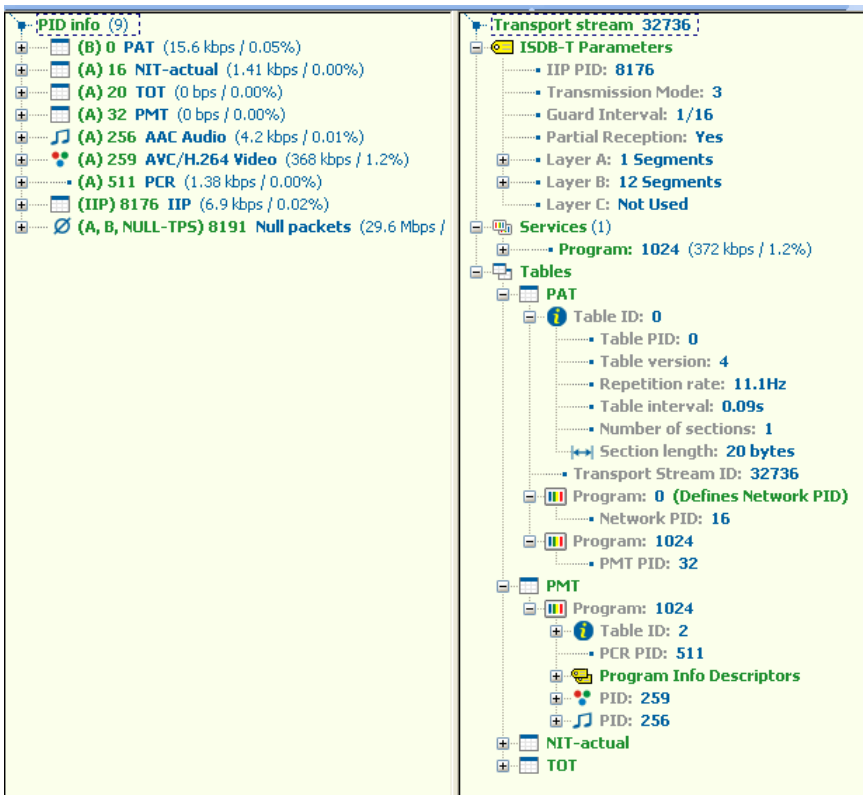


Figura 4.15: *Análise no StreamXpert da tabela PAT do fluxo de saída do multiplexador.*

4.4.2 *Buffer* de TS

O *buffer* de TS é a etapa responsável pelo armazenamento dos pacotes recebidos do TS de entrada. Ele armazena somente os primeiros 188 *bytes* de cada pacote. Essa etapa entrega em sua saída um pacote armazenado a cada solicitação de dados recebida. Essa etapa é formada por 3 sub-etapas: a memória de armazenamento dos dados, a etapa de controle de escrita e a etapa de controle de leitura da memória, conforme mostra a Figura 4.16.

A memória de armazenamento foi implementada utilizando-se uma memória FIFO (*First In First Out*) que utiliza recursos internos do próprio FPGA. A capacidade de armazenamento da memória é de 4096 *bytes*, o que equivale a

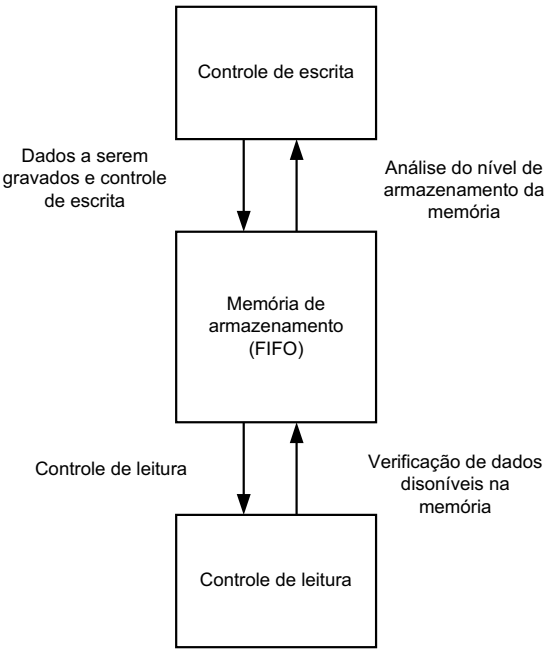


Figura 4.16: *Diagrama em blocos do buffer de TS.*

pouco mais de 21 pacotes de 188 *bytes*. O controle de escrita recebe os dados a serem armazenados. No momento de uma escrita na memória, é verificado se o espaço disponível na FIFO é suficiente para armazenar um novo pacote de 188 *bytes*. Caso não haja espaço suficiente, o pacote recebido é descartado. Já no instante da leitura, quando a memória não possui quantidade de dados suficiente para entregar um pacote completo na saída, um pacote nulo é gerado. Os dados recebidos na entrada da FIFO possuem como referência um *clock* de 27 MHz, enquanto que o *clock* de leitura é de 4,063 MHz.

4.4.3 Configurações do sistema

O multiplexador deve possuir uma interface com o usuário, pois vários parâmetros de configuração devem ser especificados, tanto com relação aos fluxos que serão transmitidos quanto com relação às configurações do transmissor que deverão ser controladas pelo multiplexador.

Deve-se especificar se o fluxo recebido em cada uma das entradas ASI será transmitido e em qual das 3 camadas hierárquicas do transmissor se dará tal transmissão. No caso da configuração do transmissor, todos os parâmetros devem ser informados pelo multiplexador: modo de operação, tempo de guarda, número de segmentos atribuídos a cada camada hierárquica, bem como as configurações associadas a cada uma dessas, como modulação, taxa de código e profundidade do entreleçador temporal.

Toda a etapa de interface com o usuário foi implementada de forma a permitir a configuração do sistema via rede. A interface faz uso de páginas WEB, criadas com a linguagem HTML, associadas a CGI's (*Common Gateway Interface*) implementados com a linguagem C. Na Figura 4.17 pode-se visualizar a página WEB através da qual se configura qual a camada de transmissão de cada TS, além da exibição de alarmes e dados do TS recebido em cada entrada. A interface WEB foi desenvolvida para 8 entradas ASI, mas o protótipo conta fisicamente com somente 4 interfaces de entrada.

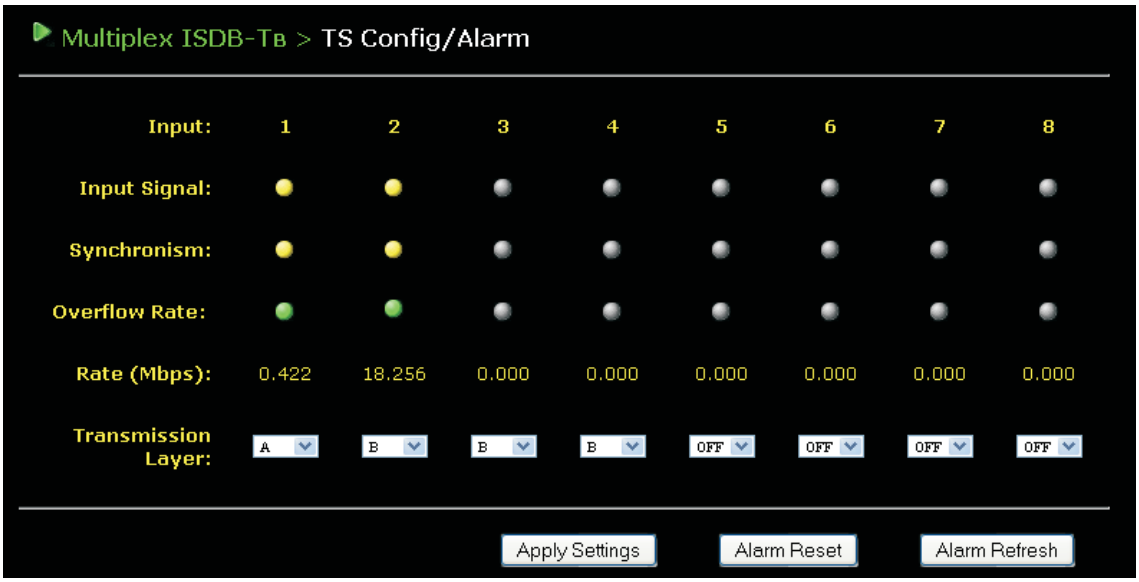


Figura 4.17: *Alarmes e configuração da camada de transmissão de cada TS.*

Na Figura 4.18 tem-se a interface de configuração dos parâmetros de transmissão.

As configurações de data e hora do multiplexador são realizadas através da interface mostrada na Figura 4.19. Esses parâmetros configuram um circuito de relógio tempo real (RTC - *Real Time Clock*) presente na placa do protótipo, que calcula automaticamente os minutos, horas, dia e ano dada uma configuração inicial. As informações extraídas do circuito de relógio tempo real são utilizadas na geração da tabela TOT.

4.4.4 Multiplexador de TS

A etapa multiplexador de TS é responsável pela multiplexação dos TS's presentes na entrada do multiplexador. A multiplexação ocorre graças a leitura dos *buffers* de TS que armazenam pacotes pertencentes a diferentes entradas ASI. O multiplexador de TS recebe solicitações da etapa controle de leitura, que repassa

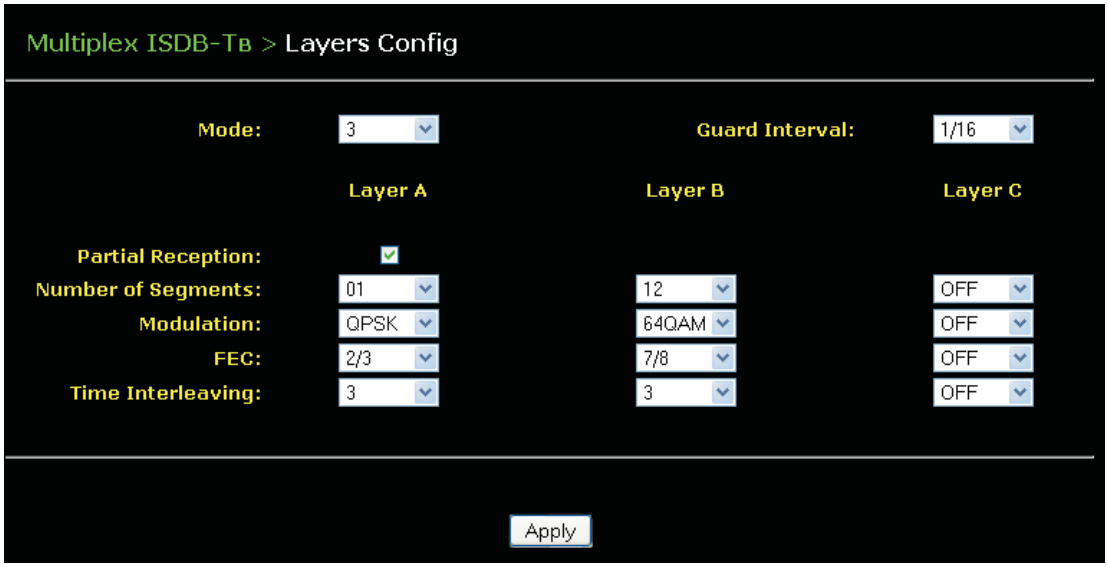


Figura 4.18: *Tela de configuração dos parâmetros do transmissão.*

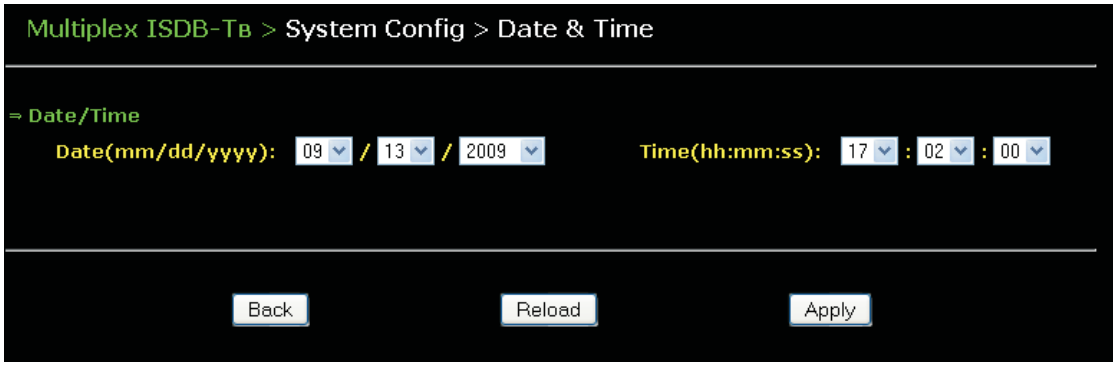


Figura 4.19: *Tela de configuração de data e hora do multiplexador.*

requisições de pacotes a serem transmitidos e informa qual será a camada hierárquica de transmissão do pacote. O multiplexador de TS monitora todos os *buffers* de entrada e tem armazenada a informação de qual é a camada hierárquica de transmissão de cada TS de entrada. A cada solicitação de leitura de pacote recebida, o multiplexador de TS verifica o nível de armazenamento dos *buffers* que contenham dados a serem transmitidos na camada em questão e lê um pacote do *buffer* que estiver com o maior nível de armazenamento naquele momento. Por exemplo, no instante em que recebe a solicitação de um pacote a ser transmitido na camada A, o multiplexador de TS analisa o nível de todos os *buffers* de TS que contém fluxos a serem transmitidos nesta camada, e realiza leitura do *buffer* que possuir maior número de pacotes armazenados. Todos os *buffers* de TS repassam dados de forma sincronizada em sua saída. Mesmo quando nenhum dado é solicitado à um *buffer* de TS, este entrega em sua saída um pacote nulo. A Figura 4.20 mostra a saída de dois *buffers* de TS. Nos sinais *out_ts1_data*, *out_ts1_sync* e

out_ts1_valid estão os sinais de saída do *buffer* de TS 1 e os sinais *out_ts2_data*, *out_ts2_sync* e *out_ts2_valid* mostram a saída do *buffer* de TS 2. Na amostra 205 da Figura 4.20 pode-se ver que o pacote repassado à saída do multiplexador de TS (sinais *mxl_data*, *mxl_out_sync* e *mxl_out_valid*) é o pacote entregue pelo *buffer* de TS 2.

Há uma comunicação entre a etapa multiplexador de TS e a etapa geração de tabelas do sistema. Quando há a solicitação de envio de alguma tabela pela etapa Geração de Tabelas do Sistema, o multiplexador de TS deixa de realizar a leitura sobre os *buffers* de TS, que passam a entregar pacotes nulos à sua saída. Neste instante, o multiplexador de TS substitui os pacotes nulos pela tabela.

4.4.5 Controle de leitura do MUX de TS

O controle de leitura dos pacotes para formação do fluxo de saída do multiplexador, que é o BTS, é realizada por esta etapa. Ela solicita ao multiplexador de TS pacotes de dados e informa, a cada solicitação, em qual camada hierárquica tal pacote será transmitido. Essas solicitações possuem uma ordem e periodicidade que são determinadas pela lei de formação do quadro de multiplexação [2][9].

O controle de leitura consiste na implementação do receptor modelo que sinaliza constantemente qual camada hierárquica está entregando um pacote na recepção, conforme mostrado na Sessão 3.3.2. Os pacotes são transmitidos pelo multiplexador na mesma ordem sinalizada pelo receptor modelo. As sinalização geradas pelo receptor modelo são utilizadas para controlar a leitura dos *buffers* de TS do multiplexador.

O processo de leitura é realizado constantemente e tem como referência um *clock* de 4,063492 MHz. Como a leitura é realizada *byte* a *byte*, a taxa resultante do processo de leitura é fixa em 32,508 Mbps.

4.4.6 Geração de tabelas do sistema

A geração das tabelas do sistema é realizada por programas implementados em linguagem C que são executados sobre o processador embarcado. O processador recebe as informações extraídas dos fluxos, fornecidas pela etapa analisador de TS, e dos parâmetros de de configuração do sistema de transmissão, fornecidas pela interface com o usuário. Tais informações são utilizadas para se criar as tabelas PAT e NIT. O processador recebe ainda informações de data e hora geradas por um circuito de relógio de tempo real. Tais informações são recebidas a cada 5 segundos e são utilizadas para se gerar a tabela TOT. Sempre que é detectada alguma alteração ou atualização nas informações usadas na geração de cada tabela, uma interrupção é enviada ao processador que deve re-gerar a tabela

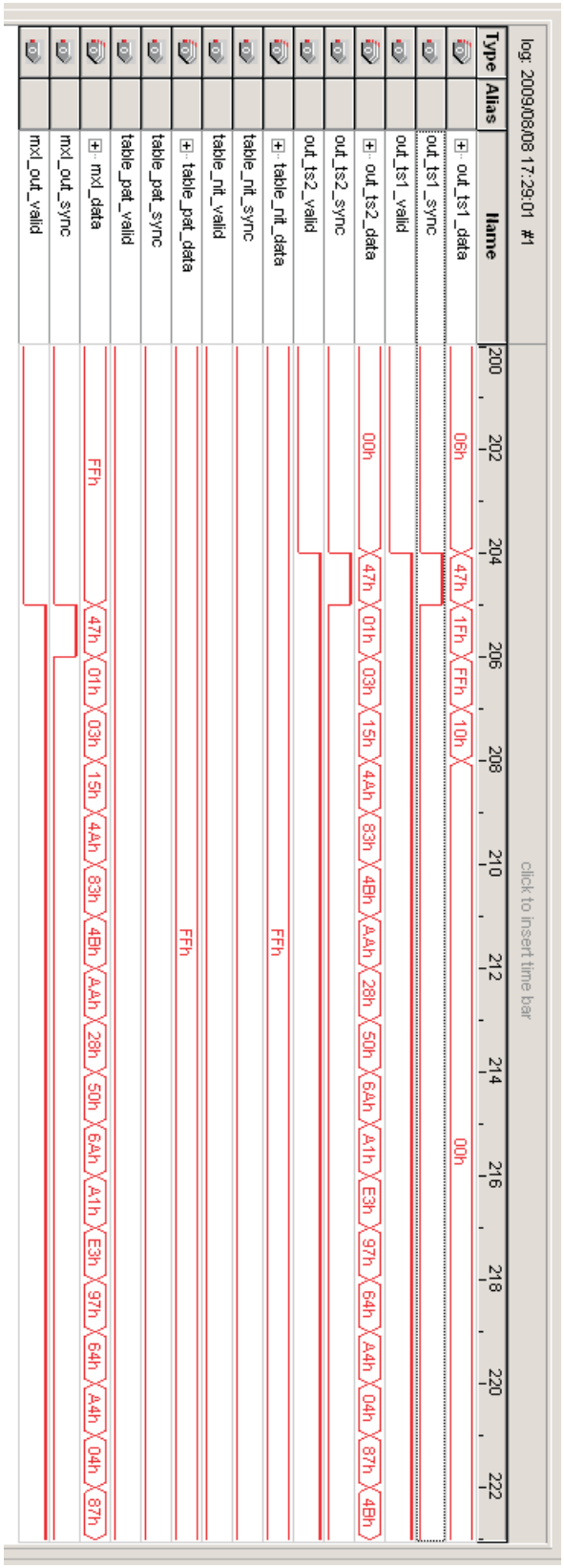


Figura 4.20: *Multiplexador de TS.*

atualizada.

As tabelas geradas são armazenadas em *buffers* externos à CPU, mas dentro do FPGA, e a cada instante de envio das tabelas esses *buffers* são lidos, não havendo necessidade de se gerar uma interrupção no processador. Isso para evitar sobrecarga do mesmo, devido a solicitação de tabelas idênticas a todo momento. A cada envio de tabela há um incremento do contador *continuity counter* presente no cabeçalho do pacote que transporta da tabela. As tabelas devem ser transmitidas no BTS respeitando um determinado intervalo de tempo, conforme exemplificado na Tabela 3.8 [7]. A PAT é enviada a cada 100 ms em média, ocorrendo o mesmo para a NIT. Já a TOT é enviada a cada 5 s em média. Um sistema de controle de envio das tabelas verifica se estes intervalos estão sendo respeitados. Caso o intervalo de tempo tenha sido atingido, o sistema substitui um pacote nulo do fluxo de saída por uma das tabelas.

A Figura 4.21 mostra uma visão dos sinais de saída de várias etapas do multiplexador, incluindo a geração de tabelas. Os sinais que tem o nome iniciado por *out_ts* são os sinais de saída do *buffer* de TS. No caso, foram exibidas os sinais referentes a dois *buffers* de TS. Os sinais que tem seu nome iniciado por *table* são de saída dos *buffers* que armazenam as tabelas para ser enviadas no fluxo de saída. Os sinais iniciados por *mxl* são os sinais de saída do multiplexador de TS. Na Figura 4.22 tem-se, no instante 0, o momento em que uma tabela PAT foi lida do *buffer* e repassada pelo multiplexador de TS às etapas seguintes.

4.4.7 Geração da IIP

Essa etapa é a responsável por gerar o IIP, um pacote de dados que controla as configurações do transmissor. O IIP é gerado com base nas informações de configuração recebidas da interface com o usuário e sua transmissão deve ser na taxa de um pacote IIP a cada quadro de multiplexação [2].

Quando o transmissor identifica que está recebendo um pacote IIP, que possui um PID especial ($0x1FF0_h$), o mesmo extrai do pacote as informações de configuração necessárias para sua operação. Essas mesmas informações são transmitidas nas portadoras TMCC, que são utilizadas pelo receptor para que este possa se configurar de maneira idêntica ao transmissor e, assim, tratar corretamente o sinal recebido. Qualquer alteração nos parâmetros de transmissão realizada no multiplexador irá se refletir em todo processo de transmissão e recepção do sinal.

Para a realização das análises mostradas a seguir, considere o protótipo configurado para o modo de transmissão 3 e intervalo de guarda de 1/16. Para essa configuração, o número de pacotes por quadro de multiplexação é de 4.352, conforme mostrado na Tabela 3.10. Na Figura 4.23 está mostrado o instante de transmissão de uma IIP. O instante de envio da IIP acontece sempre no último

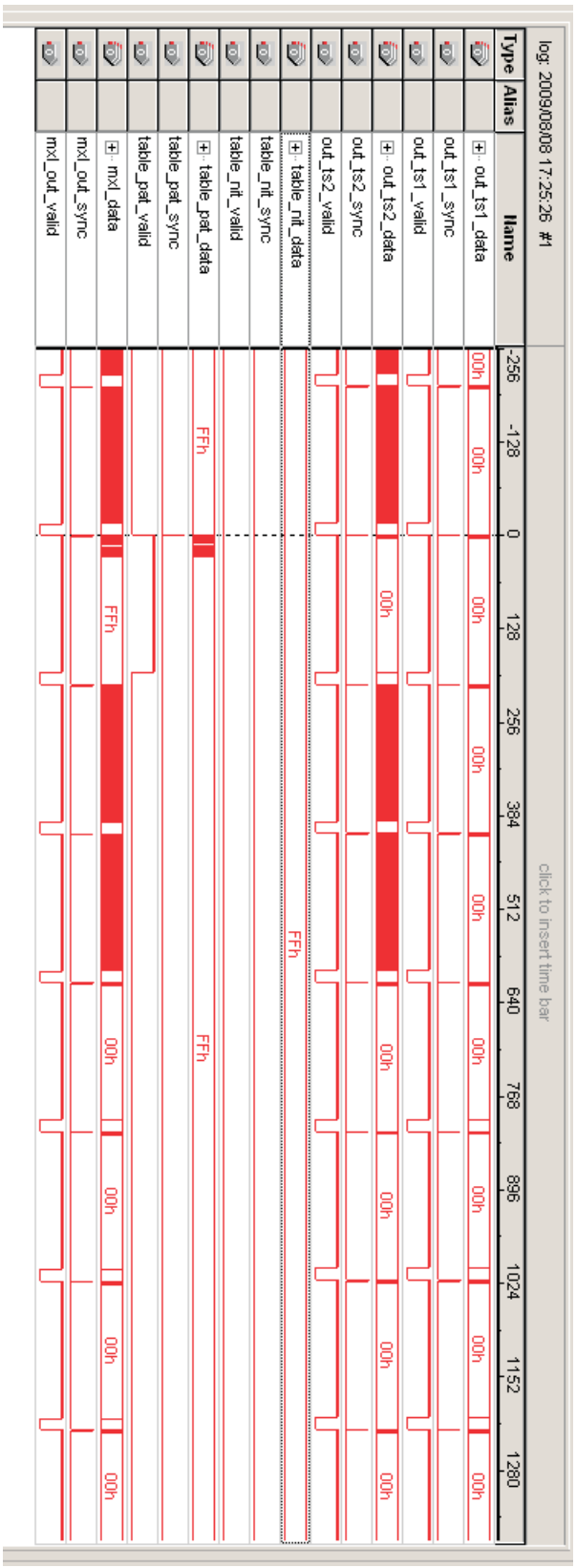


Figura 4.21: *Sinais da geração de tabela e do multiplexador de TS.*

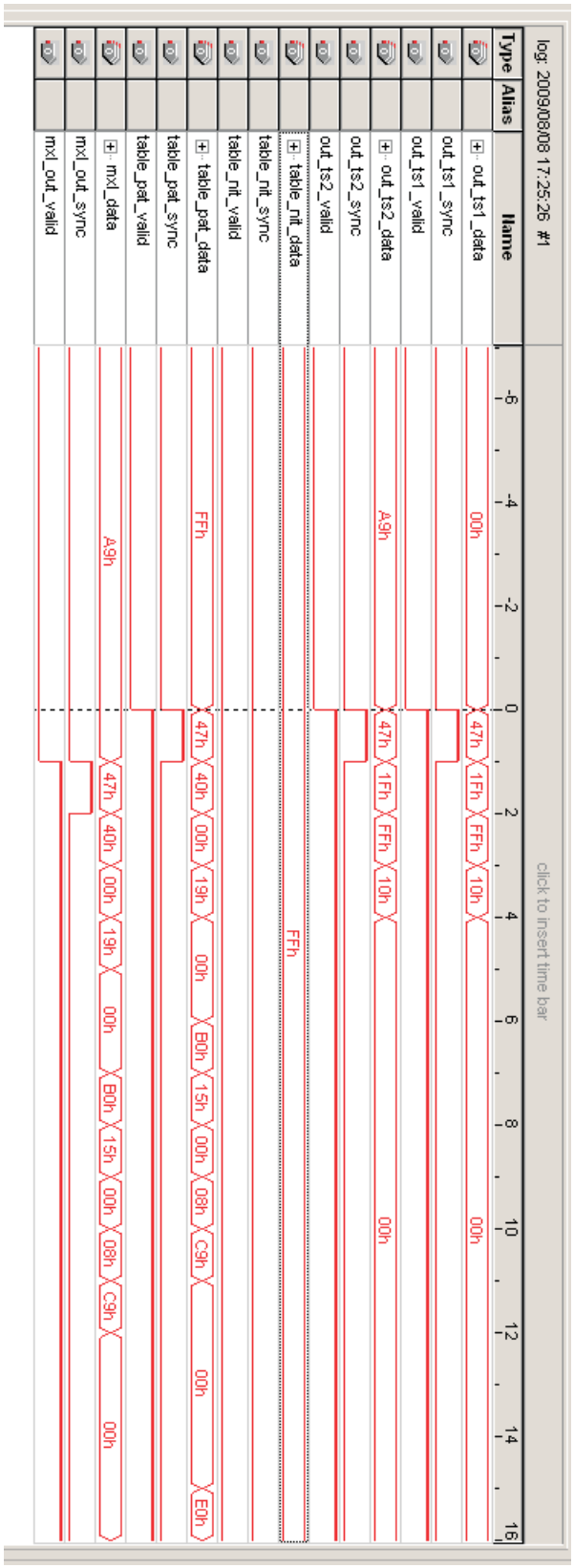


Figura 4.22: *Instante do envio da tabela PAT.*

pacote do quadro de multiplexação. Esse instante foi estabelecido para facilitar a análise da IIP, uma vez que a norma informa que uma IIP deve ser transmitida por quadro de multiplexação, mas essa não estabelece qual o instante de envio da mesma [2]. Os sinais da Figura 4.23 que iniciam seu nome por *mxl* são os sinais referentes à saída do multiplexador de TS. Pode-se destacar o sinal *mxl_tsp_counter* que indica qual o número de cada pacote dentro do quadro de multiplexação. Os sinais que possuem nome iniciado por "ii"se referem aos sinais de saída da etapa de Geração da IIP. Os dados de saída da etapa multiplexador de TS são repassados pela etapa de geração da IIP que insere, no instante certo, o pacote da IIP no fluxo de saída. Na amostra 0 da Figura 4.23 o sinal *ii_iip* assume valor 1 sinalizando o instante da inserção da IIP. O sinal *ii_tsp_counter* é um contador que indica qual é o índice do pacote dentro do quadro de multiplexação, sendo que o primeiro pacote do quadro possui índice 0. O sinal *ii_reset_frame* assume valor 1 durante o primeiro *byte* do primeiro pacote de um quadro de multiplexação. O sinal *mxl_layer_indicator* é repassado pela etapa geração da IIP e entregue em sua saída no sinal *ii_layer* e indica qual a camada hierárquica de transmissão de cada pacote do BTS sendo que $0x01_h$ indica que o pacote deve ser transmitido na camada A, $0x02_h$ na camada B, $0x03_h$ na camada C e $0x00_h$ que é um pacote nulo do quadro e deve ser descartado pelo transmissor.

A Figura 4.24 mostra o instante, na amostra de número 0, da geração da IIP. Um pacote nulo repassado pela etapa multiplexador de TS (ver *sinal mxl_data* a partir da amostra -1) foi substituído pela IIP. Isto ocorreu no último pacote do quadro de multiplexação, conforme indica *ii_tsp_counter* com valor igual a 4.351 na amostra número 0.

4.4.8 Geração dos *bytes* adicionais

Essa é a etapa responsável pela adição dos 16 *bytes* ao final de cada pacote de 188 *bytes* transmitidos. Destes 16 *bytes*, 8 carregam informações de controle enquanto os outros 8 estão previstos para paridade, conforme mostrado na Sessão 3.3.3. As principais informações transmitidas nesses *bytes* adicionais são a indicação da camada de transmissão do pacote e seu índice dentro do quadro de multiplexação, conforme mostrado na Tabela 3.11.

A Figura 4.25 mostra o término de um pacote entregue na saída da etapa de geração da IIP. Na amostra de número 187, pode-se ver no sinal *ii_dado* o último dado do pacote, que possui o valor $2D_h$. O sinal *dummy_data* repassa os dados entregues pela etapa de geração da IIP e adiciona os 16 *bytes* ao término do pacote, transformando-o em um pacote de 204 *bytes*. A partir do *byte* 189 do pacote (ver o valor $A3_h$ no instante 189 de amostragem) estão inseridos os primeiros 16 *bytes* adicionais do pacote. Os últimos 8 *bytes* destes, foram mantidos em FF_h, uma

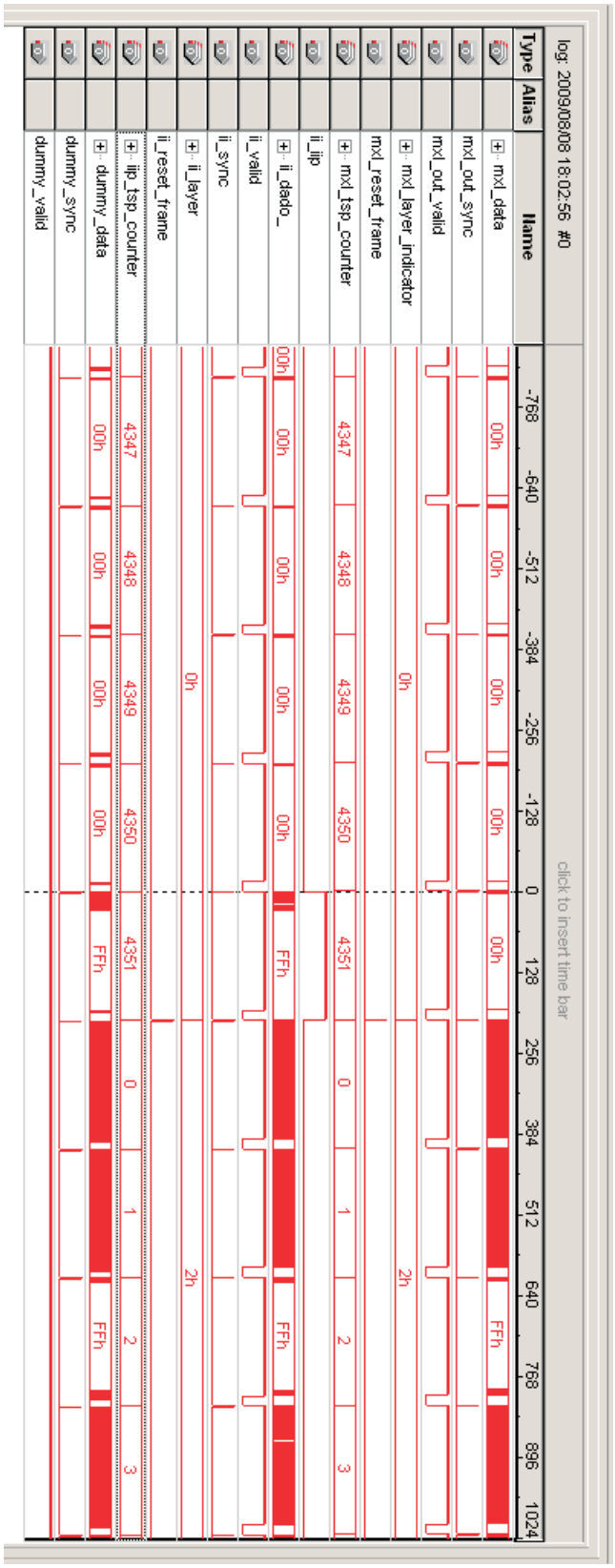


Figura 4.23: *O envio da IIP no último pacote do quadro de multiplexação.*

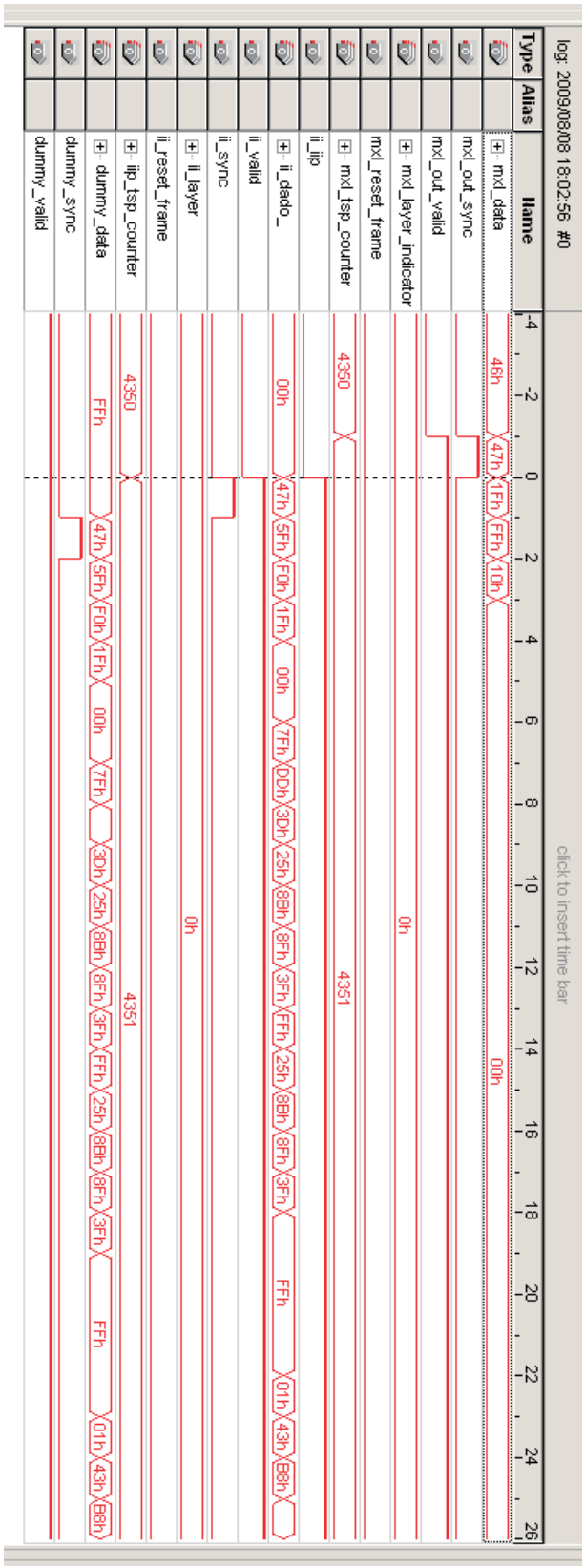


Figura 4.24: *Instante de envio da IIP.*

vez que os *bytes* de paridade opcionais não foram gerados.

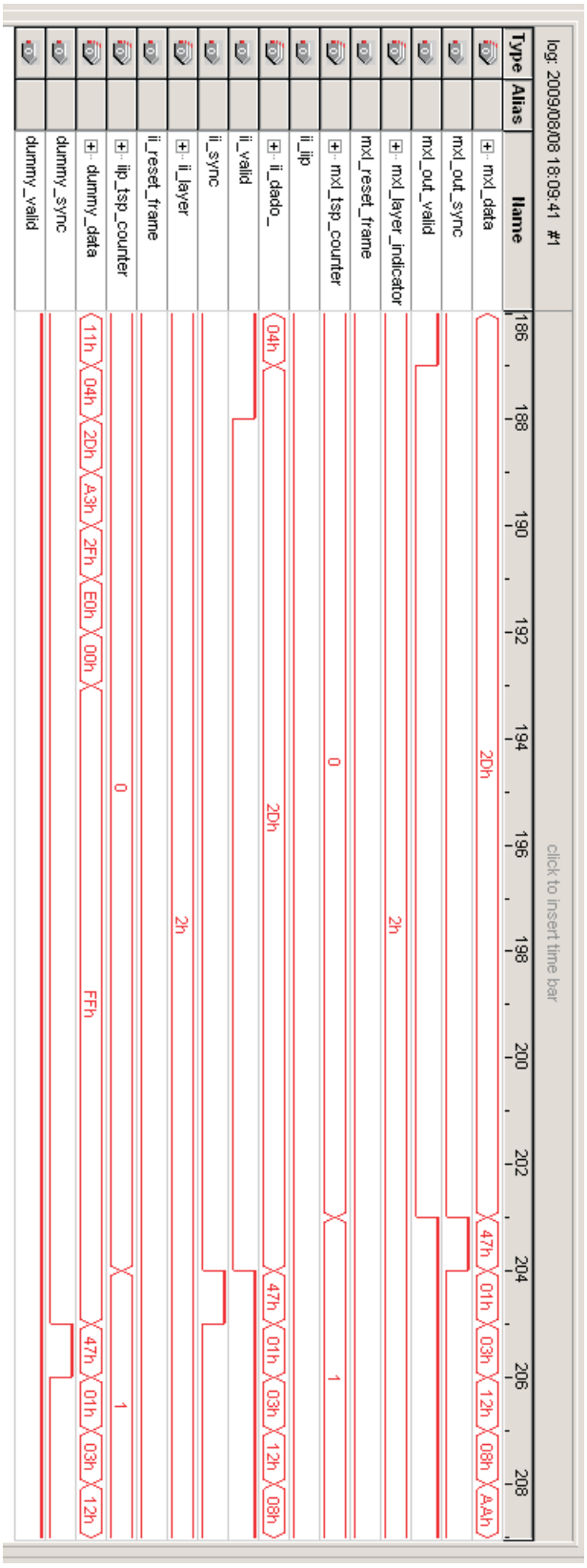


Figura 4.25: *Inserção dos 16 bytes adicionais.*

4.4.9 Tratamento do TS de saída

Essa etapa é a responsável por tratar o TS proveniente das etapas anteriores e o entregar à saída do multiplexador. O tratamento realizado envolve a mudança de domínio de *clock* dos dados, implementação da segunda fase do tratamento do PCR e entrega dos dados a placa de interface ASI. Essas tarefas são tratadas a seguir.

Mudança de domínio de *clock*

Os dados repassados pelas etapas anteriores tem como referência um *clock* de 4,063492 MHz. Esses dados são armazenados em um *buffer* de algumas dezenas de *bytes*. A leitura dos dados do *buffer* é realizada com *clock* de 27 MHz, que passa a ser a referência de *clock* dos dados.

Tratamento do PCR

Essa etapa implementa a segunda fase do tratamento do PCR apresentado na Sessão 4.4. A Figura 4.26 mostra a operação realizada nesta etapa.

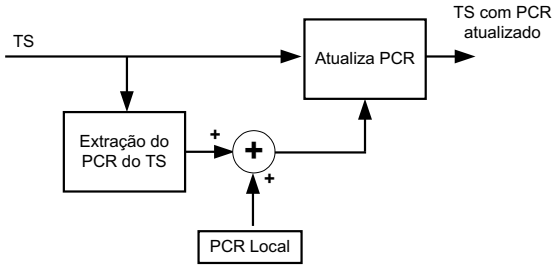


Figura 4.26: *Segunda fase do tratamento do PCR.*

Interface ASI de Saída

Os dados entregues pela etapa tratamento do PCR são repassados para a placa de circuito impresso que implementa a interface ASI de saída. A placa realiza a codificação 8B10B sobre os *bytes* recebidos, convertendo-os em 10 *bits*. Estes 10 *bits* são transmitidos serialmente ao conector de saída do multiplexador, tendo como referência um *clock* de de 270 MHz. Para manter a taxa de saída fixa, que deve ser de 270 Mbps, são gerados caracteres de preenchimento quando necessário.

4.5 Testes do protótipo

Para facilitar os testes e integração do protótipo com outros dispositivos e equipamentos, o mesmo foi montado em um *rack*, conforme mostrado na Figura 4.27. Nesse *rack*, a placa do protótipo, identificado pelo número 2 na figura, foi interligada a uma fonte de alimentação, identificada pelo número 1, e à uma placa geradora de *clock* (130,0317 MHz), indicada pelo número 3. Tanto a placa de geração de *clock* quanto a fonte de alimentação foram desenvolvidas pela empresa Linear Equipamentos Eletrônicos S/A.

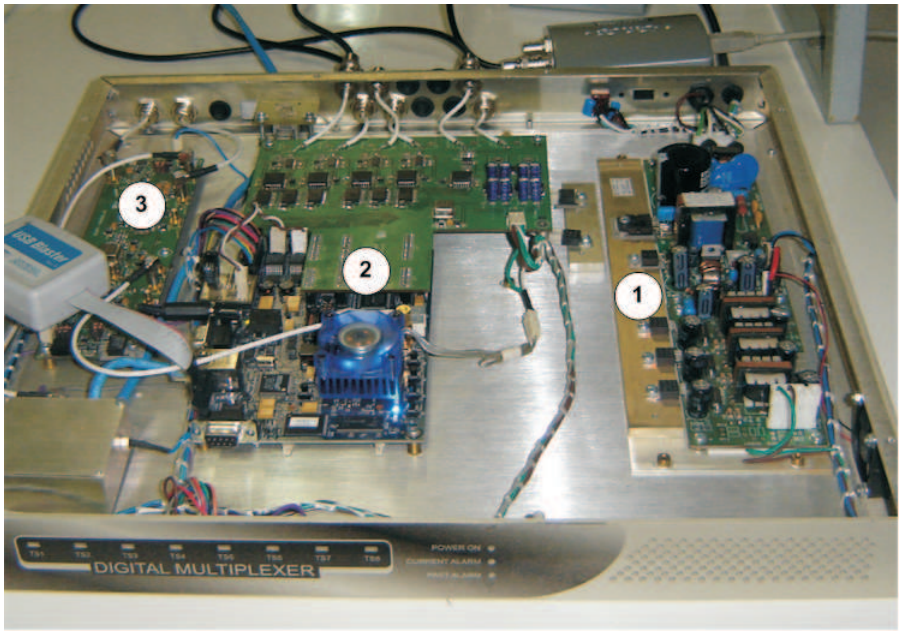


Figura 4.27: *Protótipo montado em um rack.*

A seguir estão descritos testes de validação e comprovação do correto funcionamento do multiplexador implementado. Foram realizados testes em analisadores de TS/BTS que podem indicar a integridade e conformidade do fluxo gerado.

4.5.1 Testes utilizando interfaces ASI da empresa Dektec e *softwares* de análise

Para a realização dos testes descritos a seguir, foram utilizadas duas interfaces ASI fabricadas pela empresa Dektec cujo modelo é DTU-245 FantASI [46]. Cada uma dessas interfaces oferece uma entrada e uma saída ASI que se comunica com o computador através de interface USB. Conforme mostrado na Figura 4.28, as duas saídas ASI das interfaces DTU-245 foram conectadas a duas interfaces ASI de entrada do multiplexador, para possibilitar o envio de TS's ao multiplexador a

partir do computador. Uma das entradas de uma DTU-245 foi conectada a uma saída do multiplexador para possibilitar a análise do fluxo de saída do mesmo.

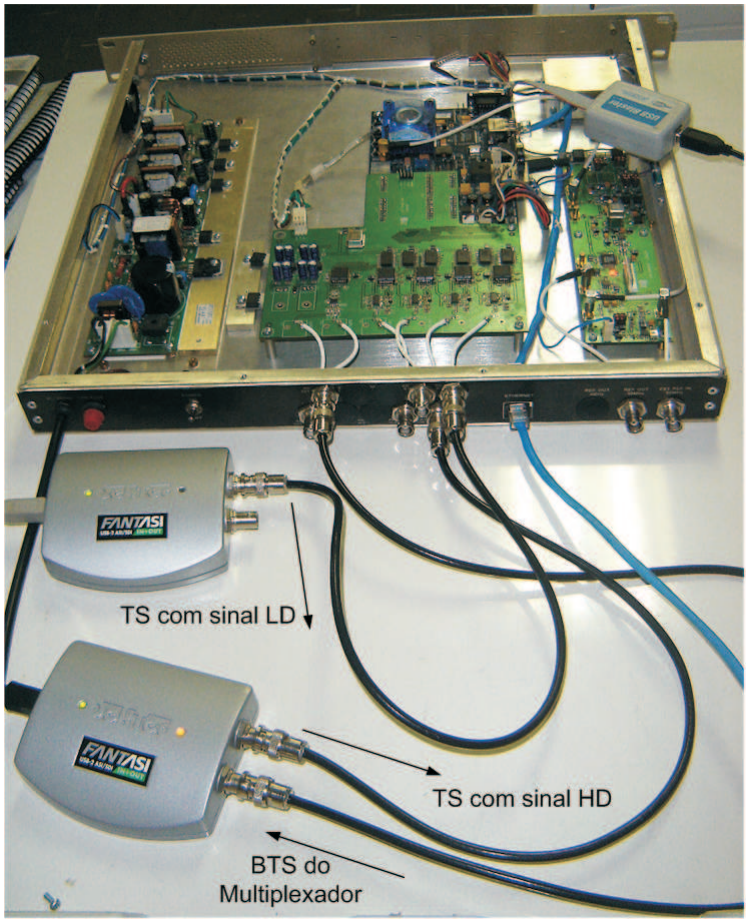


Figura 4.28: *Protótipo em teste com as interfaces Dektec.*

Dois fluxos de vídeo armazenados no computador foram reproduzidos graças a utilização das interfaces associadas ao *software StremXpress*, conforme mostra a Figura 4.30. Dois fluxos foram reproduzidos, sendo um em resolução HDTV com taxa de 12 Mbps e outro em resolução LDTV com taxa de 394,052 kbps.

No multiplexador, a entrada que recebe o fluxo LDTV foi aplicado em uma entrada ASI configurada para transmissão na camada hierárquica A, enquanto o fluxo HDTV foi aplicado em uma entrada ASI configurada para transmissão na camada hierárquica B. Os parâmetros de transmissão configurados na interface WEB são utilizados pelo multiplexador para a criação da IIP e para a correta geração do quadro de multiplexação. Nesse teste, os parâmetros de transmissão empregados são modo 3 e intervalo de guarda 1/8 e as configurações das camada hierárquicas podem ser visualizadas na Tabela 4.2.

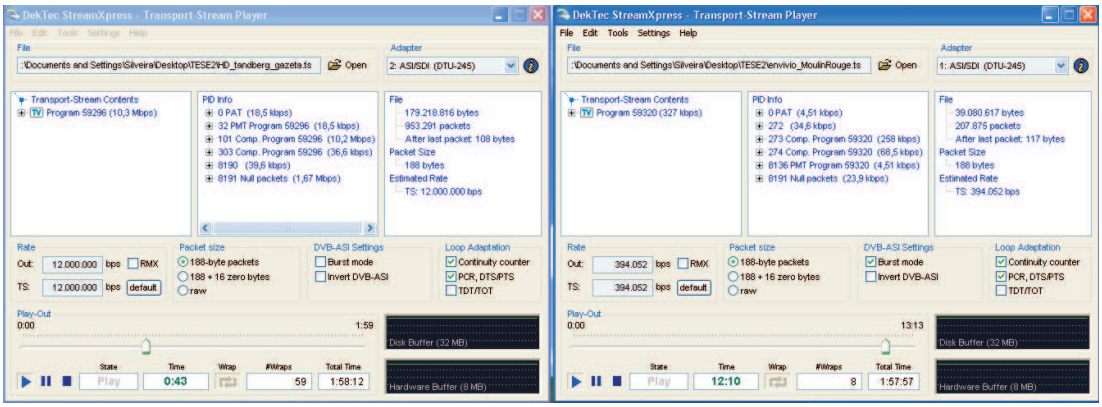


Figura 4.29: *Software StremXpress enviando fluxo às interfaces ASI.*

Tabela 4.2: *Configurações utilizadas nas camadas hierárquicas durante o teste.*

	Número de segmentos	Taxa de codificação	Modulcação	Comprimento do entrelaçador
Camada A	1	2/3	QPSK	3
Camada B	12	7/8	64QAM	3
Camada C	0	-	-	-

Análise da integridade do BTS utilizando o *software StreamXpert*

A Figura 4.30 mostra o resultado da análise em tempo real do BTS de saída do multiplexador. Embora o *software* não disponha de um perfil de análise específica para o ISDB, ele consegue analisar a estrutura de formação do TS e de tabelas e consegue identificar diversos parâmetros relacionados com o sistema ISDB e ISDB-T_B. O *software StreamXpert* é capaz de analisar os parâmetros transportados pela IIP, conforme pode ser visto no topo da parte direita da Figura 4.30. Esses parâmetros coincidem com as configurações especificados na interface WEB do multiplexador e a cada alteração dessas configurações na interface WEB acarreta em uma alteração, praticamente instantânea, nos parâmetros exibidos pelo *StremXpert*. O tempo necessário para a troca desses parâmetros, contado a partir do momento do envio dos dados pelo servidor WEB, é de 15 quadros de multiplexação, conforme comentado nas Sessões 2.2.12 e 3.3.4. A taxa do BTS gerado pode ser conferida na parte inferior da janela. O *software* indica a taxa de 29,959 Mbps em sua parte inferior. Essa indicação se deve ao fato da taxa medida considerar apenas os 188 *bytes* de cada pacote, mas convertendo a medida para pacotes de 204 *bytes* resulta em uma taxa aproximada de 32,5 Mbps, que é a taxa estabelecida para o BTS.

A data e hora configurados na interface WEB podem ser conferidas na análise da tabela TOT, conforme mostrado na parte inferior direita da Figura 4.30. As

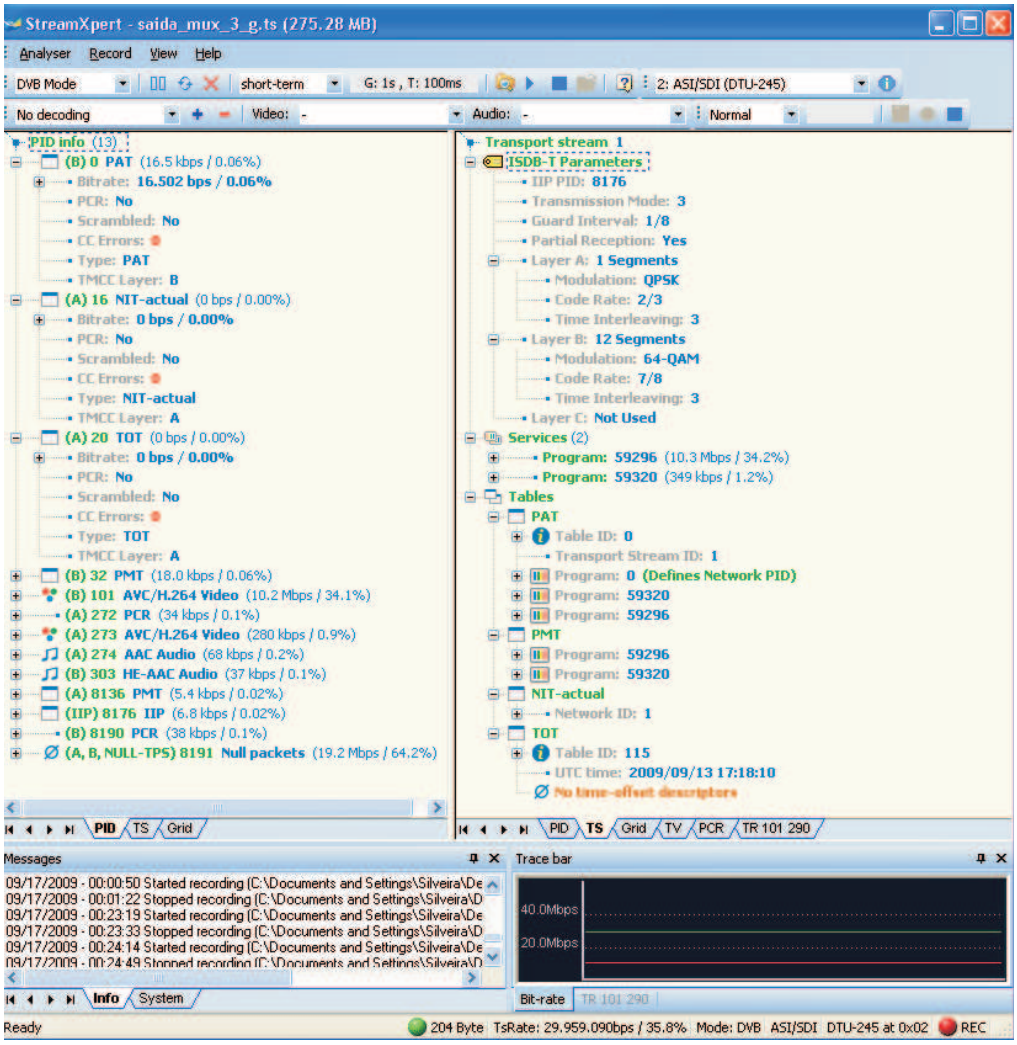


Figura 4.30: *Análise no StreamXpert do BTS de saída do multiplexador.*

tabelas PAT e NIT presentes no BTS foram também geradas pelo multiplexador de forma automática a partir do instante em que se iniciou a reprodução dos fluxos no computador. Os dois fluxos recebidos na entrada do multiplexador foram multiplexados e o *software* identificou corretamente os dois programas conforme mostra o item *services* e conforme mostra a análise da PAT na parte central do campo direito da Figura 4.30.

Na parte esquerda da Figura 4.30 são listados todos os PID's, em decimal, recebidos no BTS e, antes do valor do PID, o *software* indica qual a camada hierárquica de transmissão dos pacotes com o referido PID. Por exemplo, para a PAT, cujo PID é 0 e aparece como o primeiro PID listado na parte esquerda da janela, tem como camada de transmissão estabelecia a camada B. Na parte esquerda do *software* é possível visualizar se há erros de *continuit counter* nos pacotes de cada PID.

Existe um guia de medidas a serem realizadas com relação a integridade de

um TS, que é o *Measurement guidelines* ETR 290/TR101290 [47],[16]. Esse guia estabelece níveis de prioridades nas análises, sendo que erros nas análises de prioridade 1 indicam que o TS não será decodificado corretamente, erros nas análises de prioridade 2 indicam que o TS poderá não ser decodificado corretamente e erros de de prioridade 3 não afetam a decodificação do TS, mas sim algum informação adicional [16].

Na parte direita da Figura 4.31 estão listadas as análises conforme nível de prioridade conforme estabelece o guia a ETR 290. Conforme mostra a Figura 4.31, a análise do BTS não apresentou erros de prioridade 1. Ou seja, não ocorrem erros com relação ao sincronismo do TS (TS_*sync_loss*) ou dos pacotes (*Sync_byte_error*) e de formação das tabelas essenciais do TS, PAT (PAT_*error*) e PMT (PMT_*error*). Também não aconteceram erros de *continuity counter* (*Continuit_count_error*), indicando que durante o processo de multiplexação foram mantidas as ordem dos pacotes de cada TS e que não foram perdidos pacotes durante armazenamento nos *buffers* internos do multiplexador. Erros relacionados a PID's (PID_*error*) encontrados mas não identificados ou por PID's identificados mas não localizados no TS também não ocorreram.

Com relação a erros de prioridade 2, foi verificada a ausência de erros durante o processo de transmissão dos pacotes (*Transport_error*), através da análise do campo *Transport error indicator* descrito na Sessão 3.1.2. Erros de formação dos pacotes, (CRC_*error*) também não foram encontrados através da verificação do CRC dos pacotes. Erros de intervalo de repetição dos pacotes com PCR foram acusados (PCR_*repetition_error*). O *software* indica que o intervalo entre pacotes com PCR está acima de 40 ms. Porém, esse é o intervalo estabelecido para o sistema DVB [16]. No caso do sistema ISDB, a norma indica que esse valor pode chegar próximo a 65 ms [2]. Portanto, pode-se desconsiderar esse erro, uma vez que ele indica a falta de acordo com o padrão DVB. Caso haja diferença de valores maiores que 100 ms entre sucessivos PCR's e indicações de descontinuidades não tenham sido indicadas no *adaptation field*, serão indicados erros em PCR_*discontinuity_indicator_error*, o que não aconteceu. Erros com relação a precisão do PCR (PCR_*accuracy*) também não foram encontrados. Isso indica que o processo de atualização de PCR implementado, mostrado na Sessão 4.4.1, conseguiu atender as especificação da norma. Erros com relação a Acesso condicional (CAT_*error*) não foram indicados, até mesmo por que não está sendo realizada a transmissão da tabela CAT e o TS não possui restrição de acesso.

Com relação aos erros de prioridade 3, não foram localizados erros com relação a tabela NIT (NT_*actual_error* e NIT_*other_error*). Os erros de prioridade 3 indicados se referem as tabelas EIT, SDT e TDT que realmente não estão sendo geradas pelo protótipo.

Na Figura 4.32 estão mostradas análises do PCR com relação ao fluxo LD

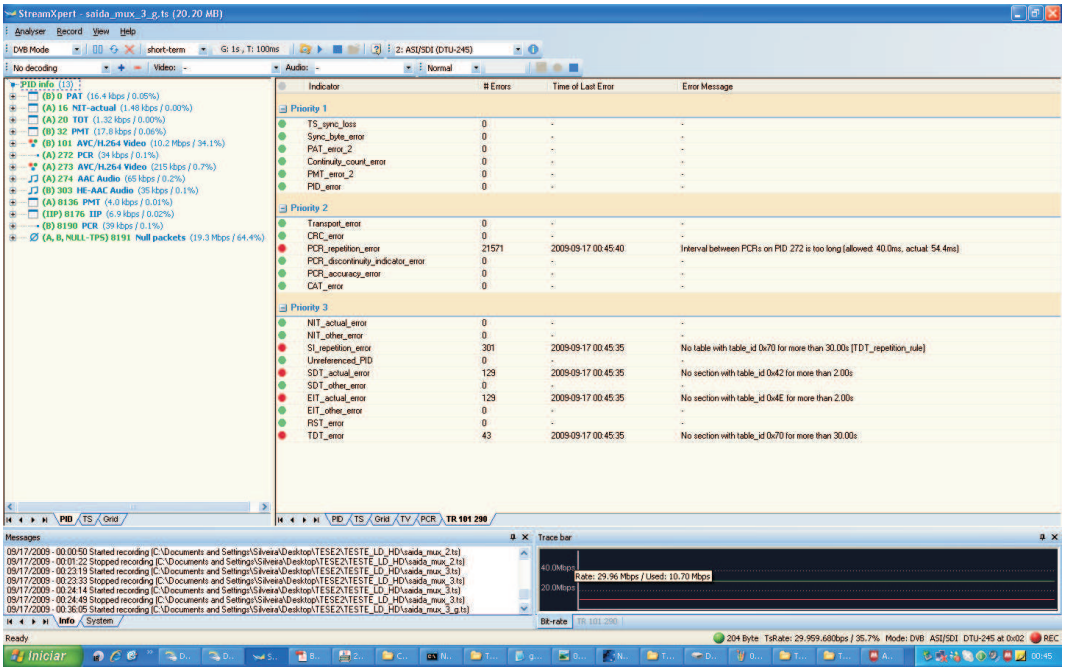


Figura 4.31: *Análise de erros pelo StreamXpert segundo a ETR 290/TR101290.*

presente na saída do multiplexador. A medida intitulada de PCR_AC *Snapshot*, mostra que a imprecisão dos valores do PCR (*jitter*) a cada pacote ao longo do tempo. A medida PCR_AC mostra a distribuição acumulada da imprecisão de PCR, indicando que a maior parte das imprecisões acontecem com valores abaixo de (+/-)80 a (+/-) 100 ns, sendo que o máximo estabelecido por norma é de 500 ns [10]. A medida PCR_interval indica a distribuição dos intervalos de tempo entre os pacotes com PCR e mostra que a maioria dos pacotes com PCR estão espaçados de 30 a 50 ms.

As análises do PCR com relação ao fluxo HD presente na saída do multiplexador, estão mostradas na Figura 4.33. As medidas mostram que a distribuição acumulada da imprecisão do PCR, PCR_AC, indicando que a maior parte das imprecisões acontecem também com valores em torno de (+/-) 80 ns e a medida PCR_interval indica a distribuição dos intervalos de tempo entre os pacotes com PCR, indicando que a maioria dos pacotes com PCR se concentra entre os valores de 30 e 40 ms.

Além das análises do fluxo mostradas anteriormente, o *StreamXpress* também é capaz de decodificar os programas recebidos. Nas Figuras 4.34 e 4.35 estão mostrados respectivamentes os vídeos LD e HD sendo decodificados a partir do BTS do multiplexador. Isto comprova que os fluxos multiplexados são interpretados e decodificados com sucesso pelo *software*.

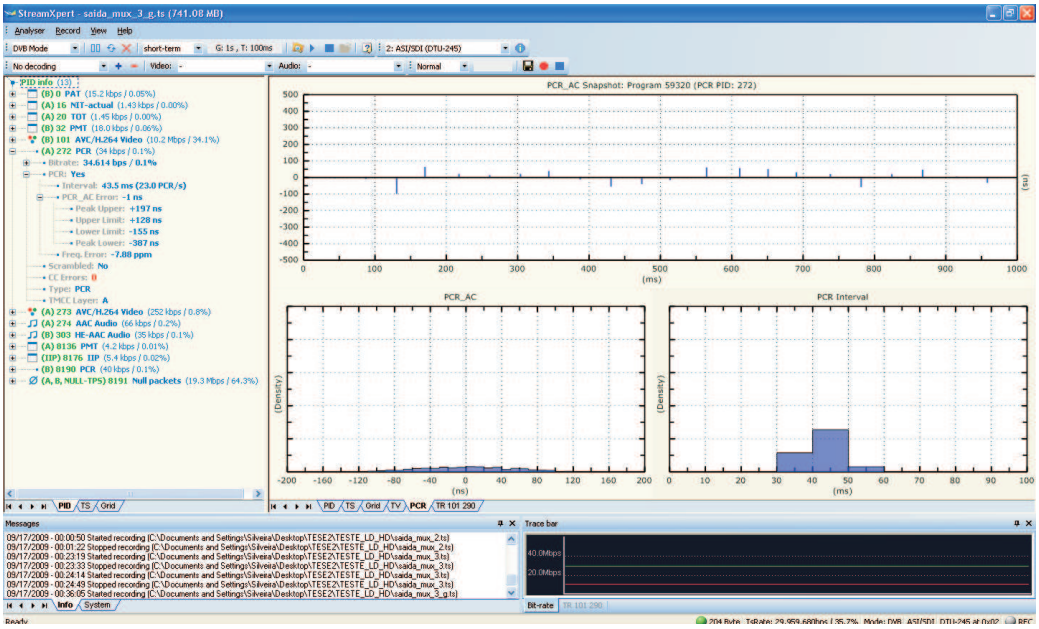


Figura 4.32: *Análise de PCR do fluxo LD no StreamXpert.*

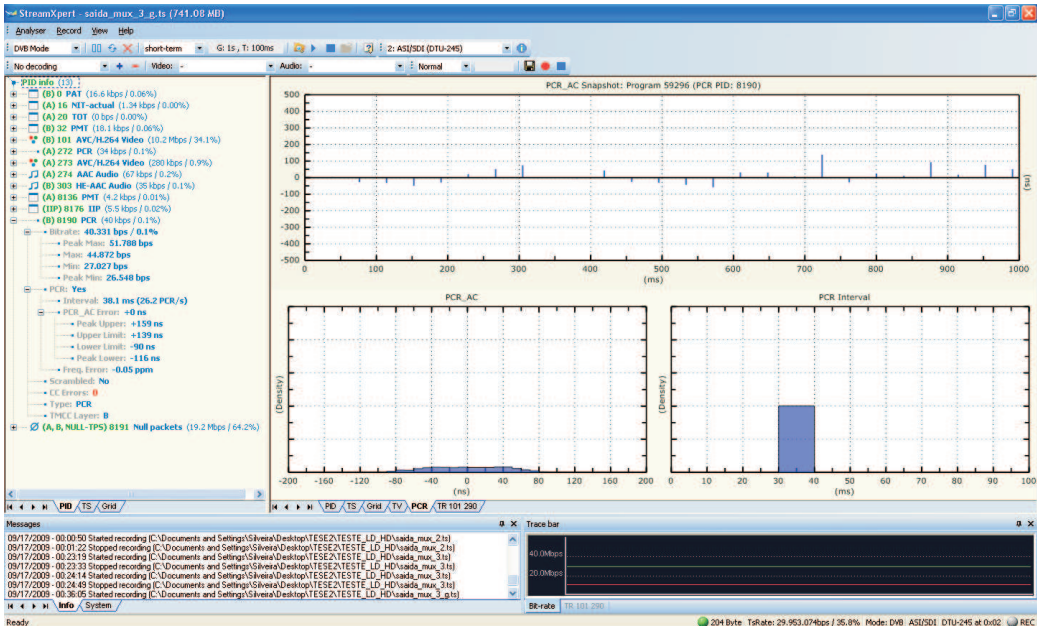


Figura 4.33: *Análise de PCR do fluxo HD no StreamXpert.*

Análise da integridade do BTS utilizando o MPEG TS *Compliance Analyzer*

Através do *software StreamXpert* é possível analisar e decodificar o BTS de saída do multiplexador, conforme mostrado anteriormente. Utilizando este mesmo *software*, foi gravado um trecho do BTS de saída do multiplexador para a realização de análises em um segundo *software*, o MPEG TS *Compliance Analyzer*[18], cu-

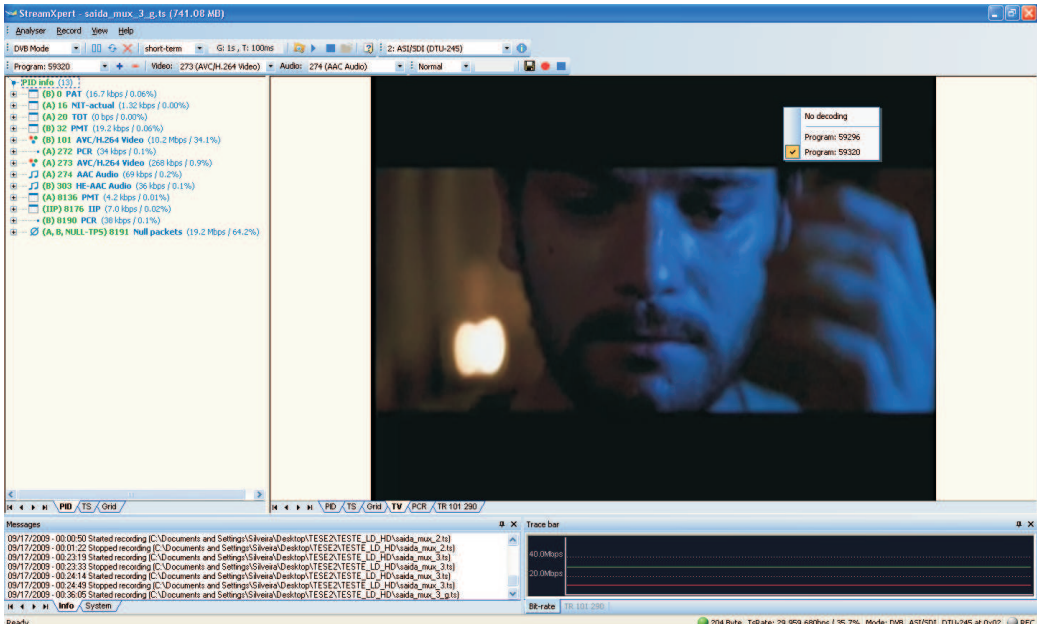


Figura 4.34: *Decodificação do fluxo LD no StreamXpert.*

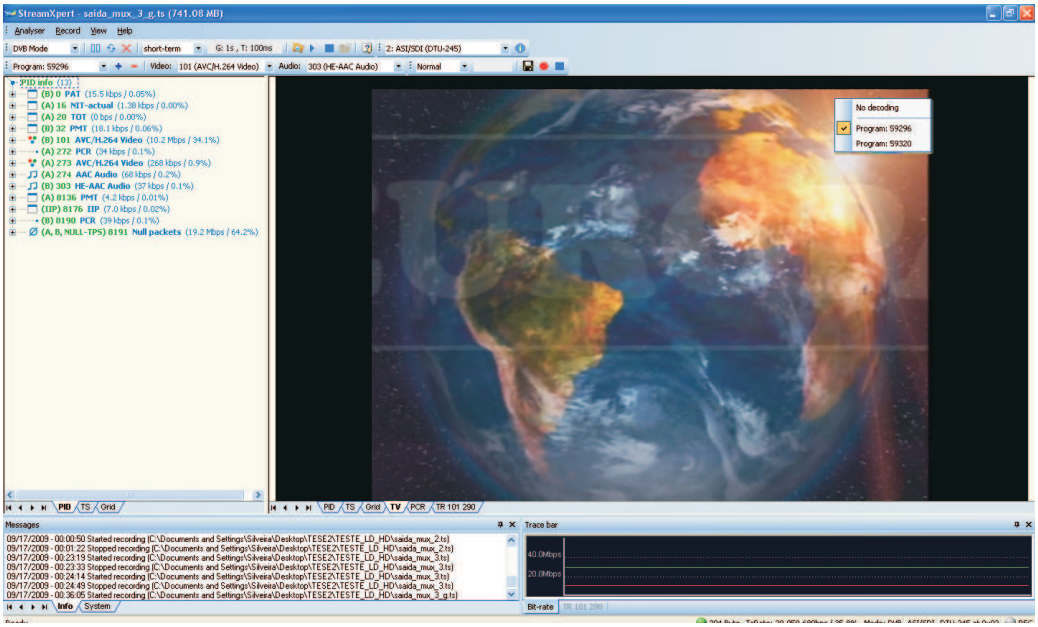


Figura 4.35: *Decodificação do fluxo HD no StreamXpert.*

jos resultados serão mostrados a seguir. Esse *software* possibilita a análise de TS/BTS de maneira similar ao *StremXpert* e realiza verificações com relação a parâmetros específicos do sistema ISDB. Ele realiza somente a análise de fluxos gravados, ao contrário do *StremXpert* que realiza a análise em tempo real do fluxo recebido na interface ASI conectada ao computador.

A Figura 4.36 mostra uma tela de análise no MPEG TS *Compliance Analyzer*

do fluxo da saída do multiplexador. No lado superior esquerdo da figura estão mostrados que os programas transmitidos no fluxo de saída do multipelxador foram corretamente interpretados pelo *software*. Do lado direito estão os resultados de alguns testes e validações com relação a parâmetros do sistema ISDB. Nenhum erro foi encontrado. Esses testes verificam principalmente a consistência das informações de transmissão e formação da tabela NIT, considerando os descritores específicos do sistema ISDB que ela transporta.

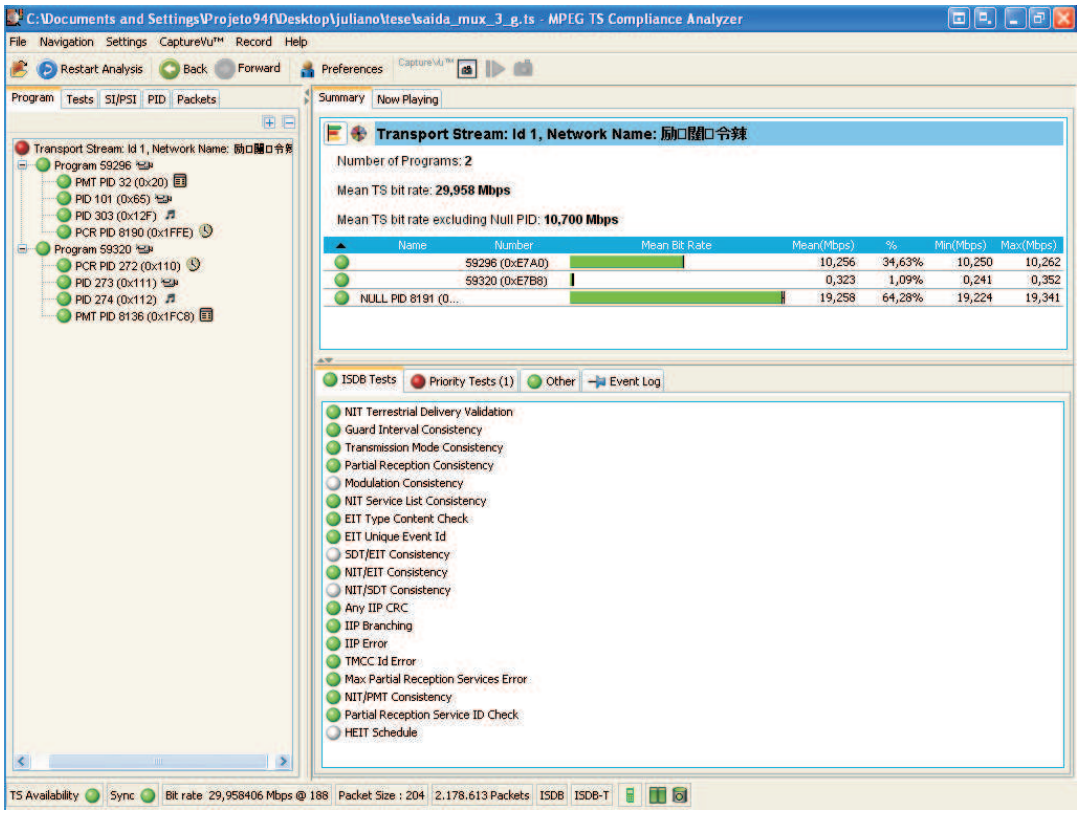


Figura 4.36: *Análise do BTS no MPEG TS Compliance Analyzer.*

O *software* MPEG TS *Compliance Analyzer* também realiza teste com relação a consistência do TS/BTS baseados no guia ETR 290/TR101290. A Figura 4.37 mostra que as análises realizadas são idênticas as realizadas pelo *StremXpert* e indicam apenas erro com relação a tabela SDT que não está sendo enviada pelo protótipo.

As Figuras 4.38 e 4.39 mostram, respectivamente, as análise com relação ao PCR dos programs LD e HD presentes no BTS de saída do multiplexador. Conforme mostrada na parte inferior direita da figura 4.38, a imprecisão de PCR do fluxo LD se concentra abaixo de (+/-) 100 ns e espaçamento entre pacotes com PCR, mostrado na parte superior do lado direito da mesma figura, se concentra entre 30 e 50 ms. Esses dados coincidem com a análise realizada no *StreamXpert*.

A análise do fluxo HD, Figura 4.39, também coincidem com as mostradas pelo

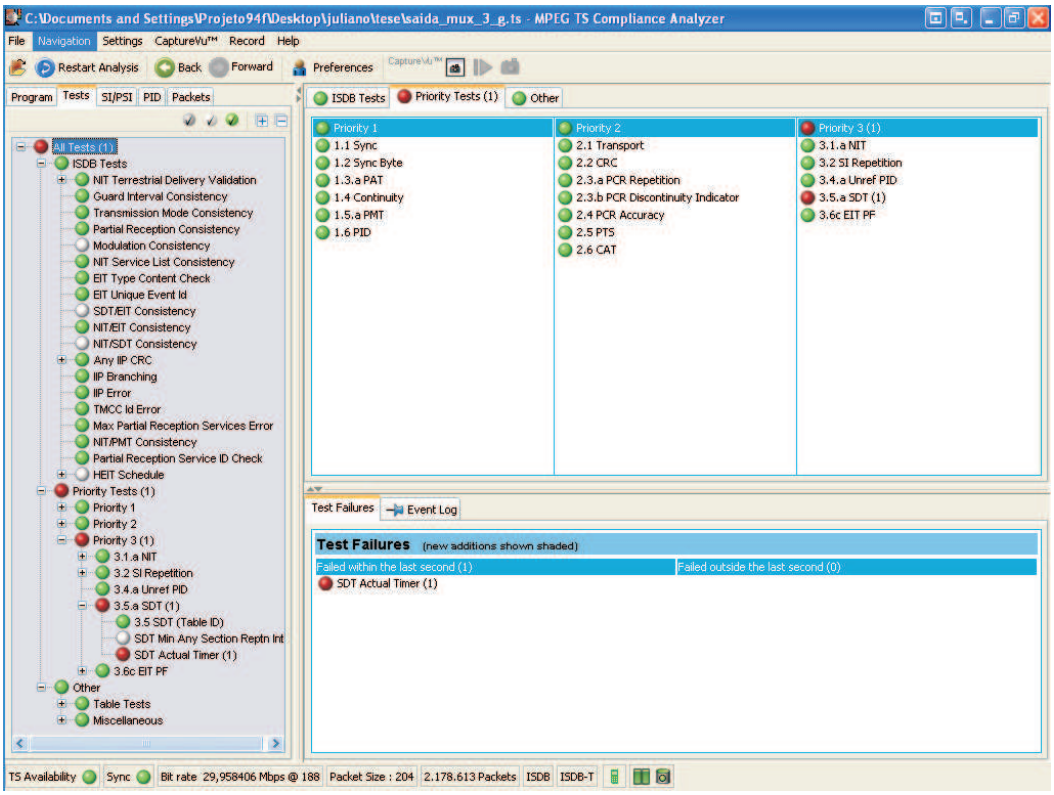


Figura 4.37: *Verificação de erros do BTS pelo MPEG TS Compliance Analyzer.*

StremXpert. A imprecisão do PCR concentrada em (+/-) 100 ns e a média dos intervalos de tempo entre os pacotes com PCR se concentra próxima a 40 ms, conforme indicado também pelo *StremXpert*.

As informações de data e hora interpretadas a partir da tabela TOT pode ser visualizadas na Figura 4.40.

4.5.2 Teste utilizando equipamentos profissionais de transmissão

A seguir está mostrado o resultado de teste do protótipo do multiplexador realizado em conjunto com equipamentos profissionais de transmissão de TV digital. Neste teste foram aplicados 3 TS's na entrada do multiplexador, conforme mostrado no diagrama da Figura 4.41. Dois TS's foram obtidos na saída de codificadores de vídeo, sendo um codificador de vídeo HD e outro LD, e um terceiro TS foi obtido na saída de uma interface ASI da Dektec, que reproduzia um TS SD. Uma das saídas do protótipo foi ligada à interface ASI da Dektec, para análise do fluxo de saída do multiplexador através do *software StreamXpert*, e a outra saída foi conectada em um transmissor profissional do padrão ISDB-T_B, para a realização de teste de transmissão do BTS. Para a recepção do sinal irradiado

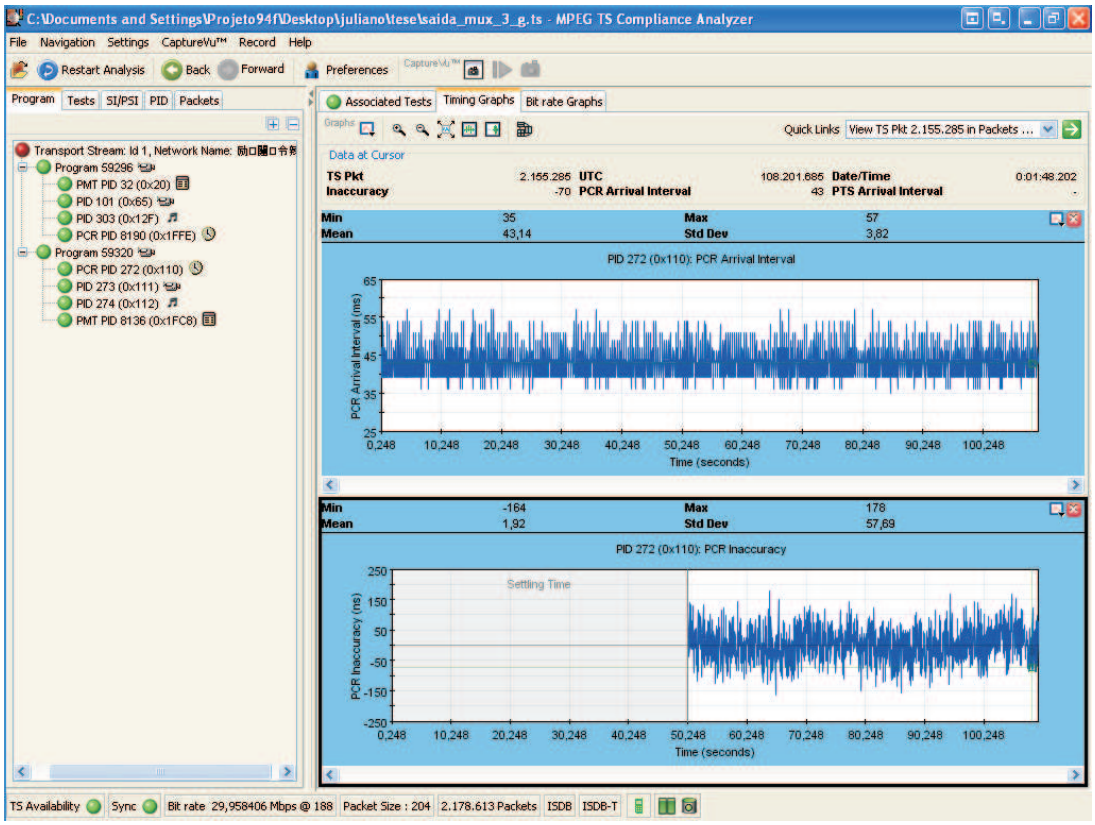


Figura 4.38: *Análise do PCR do programa LD pelo MPEG TS Compliance Analyzer.*

pelo transmissor foram utilizados um receptor fixo e um receptor móvel.

A Figura 4.42 mostra os equipamentos envolvidos no teste. O equipamento identificado pelo número 1 na figura entrega o vídeo sem compressão ao codificador de vídeo HD, indicado pelo número 4. O equipamento número 2 entrega o sinal de vídeo sem compressão ao codificador LD, identificado pelo número 6. Os codificadores de vídeo, por sua vez, entregam o sinal comprimido e encapsulado em um TS ao protótipo do multiplexador identificado pelo número 3. E este, entrega o BTS de saída ao modulador do padrão ISDB-T_B que está identificado pelo número 5 na Figura 4.42. O modulador entrega o sinal já em canal em sua saída conectada à uma antena. A Tabela 4.3 identifica cada um dos equipamentos utilizados neste teste. Na mesma figura, é possível visualizar o receptor móvel, conectado ao computador, indicado pelo número 7. Os parâmetros de transmissão utilizados são os mesmo do teste mostrado na Sessão 4.5.1. Na sequência estão mostradas as análises realizadas durante o teste.

Análise do BTS utilizando o *software StreamXpert*

Estão mostradas a seguir o resultado da análise do fluxo de saída do multiplexador no *software StreamXpert*. Ao observar o item *services* e observar a análise da

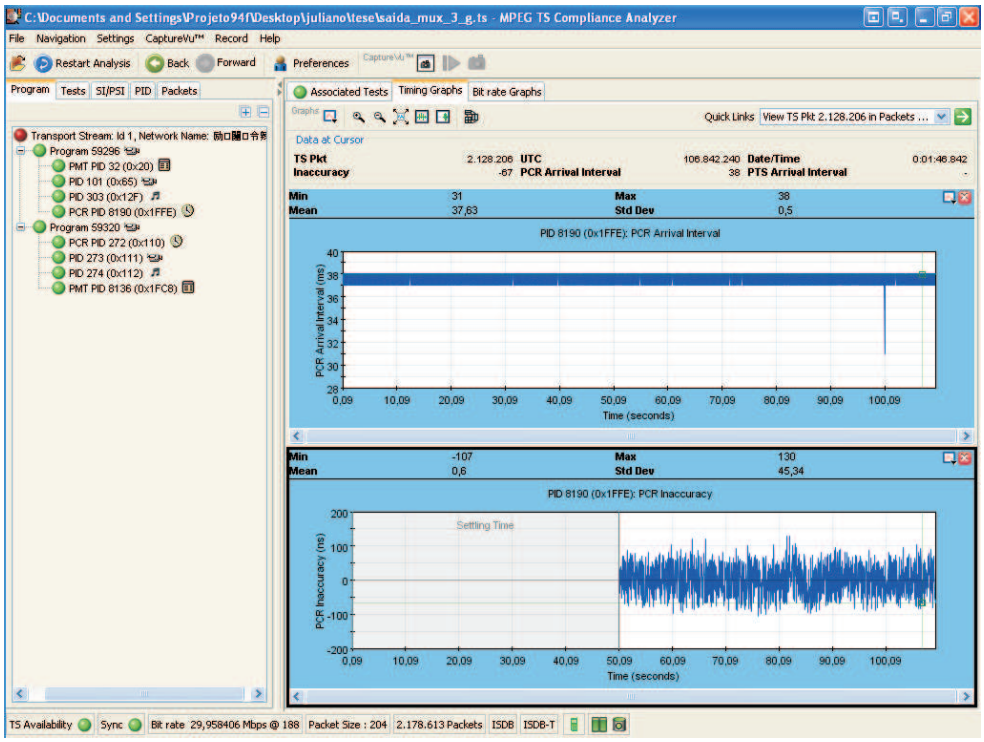


Figura 4.39: *Análise do PCR do programa HD pelo MPEG TS Compliance Analyzer.*

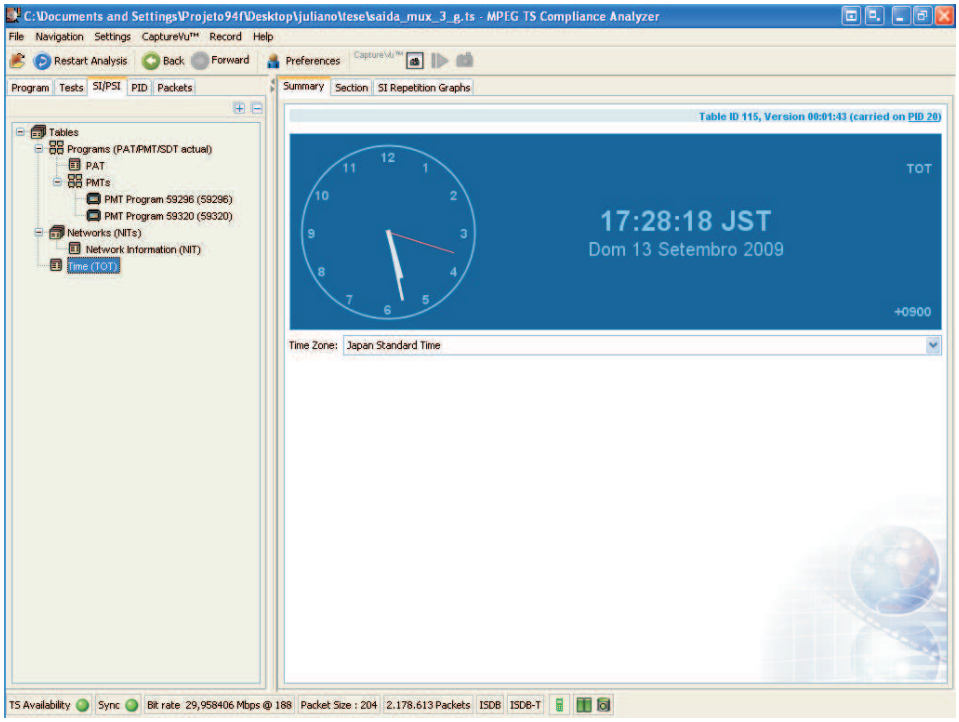


Figura 4.40: *Informações de data e hora extraídas da TOT.*

PAT, no lado direito da Figura 4.43, é possível visualizar que foram identificados três programas no BTS. Os três programas multiplexados , LD, SD e HD, estão

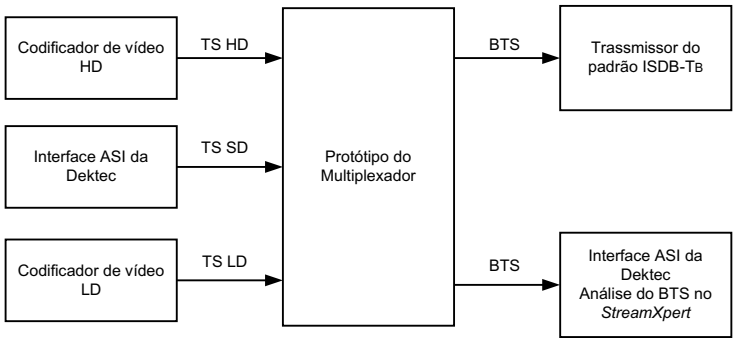


Figura 4.41: *Diagrama de transmissão utilizado no teste.*

Tabela 4.3: *Identificação dos equipamentos utilizados no Teste.*

Número	Equipamento	Fabricante	Descrição
1	V1HD/LE [48]	Doremi Labs Inc.	Servidor de vídeo HD
2	IRD3384A [49]	Sencore	IRD ATSC e Decoder MPEG-2
3	Multiplexador	—-	Protótipo sob teste
4	Kyrion Ah2101 [50]	Ateme/Linear	Codificador de vídeo HD
5	IS7001 [51]	Linear	Modulador ISDB-TB
6	OTT7700 [52]	onTimeTek Inc.	Codificador de vídeo LD

identificados no no item *services* da figura. O programa 1, de aproximadamente 10 Mbps é o vídeo HD, o programa 56, de aproximadamente 340 kbps, é o vídeo LD e o programa 51648, de aproximadamente 2.4 Mbps é o vídeo SD. As configurações de transmissão configuradas na interface WEB estão corretamente mostradas, na parte superior direita da Figura 4.43, na análise da IIP do BTS.

A Figura 4.44 mostra a análise de integridade do BTS de saída do multiplexador. Pode-se observar que não foram encontrados erros de prioridade 1 e que o único erro de prioridade 2 apontado é com relação o de intervalo de repetição dos pacotes com PCR (PCR_*repetition_error*). Porém, conforme já comentado, o *software* indica que o intervalo entre pacotes com PCR está acima de 40 ms e isto é aceitável no padrão ISDB [2]. Erros de prioridade 3 mostram que não ocorreram erros com relação a tabela NIT (NT_*actual_error* e NIT_*other_error*). Os erros de prioridade 3 indicados se referem as tabelas EIT, SDT e TDT que não estão sendo geradas pelo protótipo. Estas análises indicam que o processo de multiplexação dos 3 TS's pelo multiplexador foi realizada de forma correta.

Nas Figuras 4.45, 4.46 e 4.47 estão mostradas, respectivamente, as análises de PCR dos fluxos LD, SD e HD do BTS. As três análises mostram, através da medida PCR_AC na parte inferior direita de cada figura, que a maior parte das imprecisões de PCR acontecem com valores em torno de (+/-) 100 ns o que



Figura 4.42: *Equipamentos de transmissão utilizados no teste.*

atende a recomendação da norma do MPEG que é de (+/-) 500 ns.

Nas Figuras 4.48, 4.49 e 4.50 estão mostradas a decodificação de cada fluxo do BTS no *software StreamXpert*, comprovando que os três fluxos multiplexados, LD,

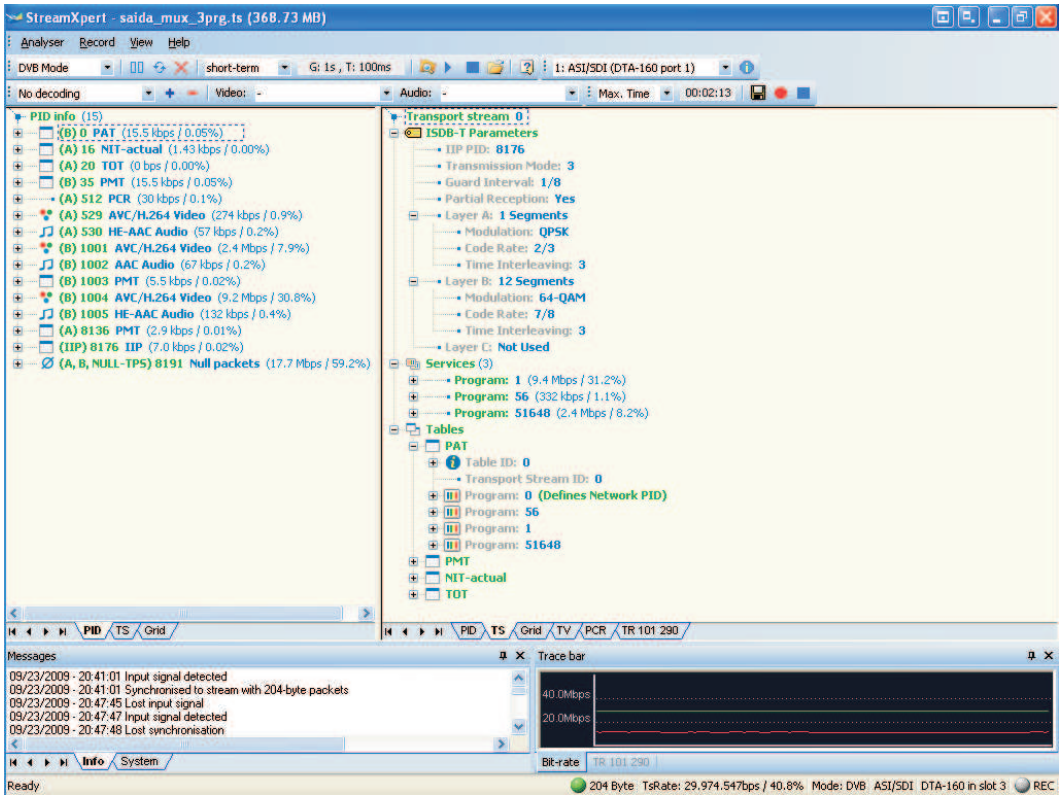


Figura 4.43: *Análise do BTS de saída do multiplexador.*

SD e HD respectivamente, puderam ser corretamente identificados e decodificados pelo *software*.

Transmissão e recepção do BTS

A saída de BTS do multiplexador foi interligada à entrada ASI de um modulador do padrão ISDB-T_B. Na Figura 4.51 está mostrada parte frontal do modulador, onde está mostrada a interface de visualização dos parâmetros de transmissão do mesmo. A interface mostra que o modulador está configurado conforme os parâmetros extraídos de um BTS, conforme indica a primeira linha do visor do equipamento: "*Curreent TMCC Monitor: BTS IN*". Os parâmetros que foram extraídos do BTS, mostrados na mesma figura, coincidem com os parâmetros transmitidos na IIP do BTS. Isto indica que a IIP gerada pelo multiplexador foi corretamente interpretada pelo modulador. Do lado direito da Figura 4.51 pode-se visualizar, através dos led's de sinalização de alarmes, que não foram acusados nenhum tipo de problema no funcionamento do modulador. Pode-se concluir que o BTS gerado pelo multiplexador foi corretamente recebido e interpretado pelo modulador.

A recepção do sinal transmitido pelo modulador foi realizada por um receptor

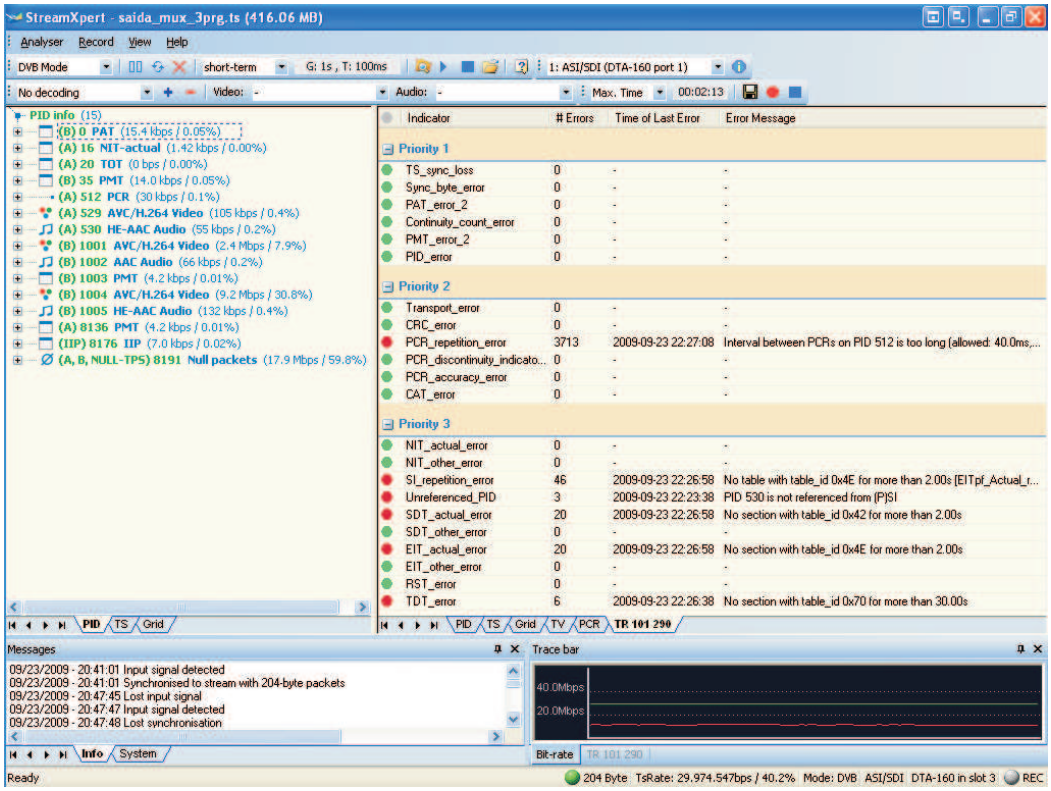


Figura 4.44: *Análise de integridade do BTS de saída do multiplexador.*

fixo da empresa *Zinwell* (modelo ZBT620)[53] e por um receptor móvel, do tipo USB (*Universal Serial Bus*), da empresa SEMP TOSHIBA (modelo DT1020) [54]. Na Figura 4.52 está mostrada uma foto da tela do computador onde foi instalado o receptor móvel. O receptor móvel capta o sinal via antena e exibe o sinal decodificado em seu *software*, mostrado na parte inferior esquerda da figura. Na mesma figura é possível visualizar o *software StremXpert* que realiza a análise do BTS de saída do multiplexador. Ambos reconhecem e decodificam perfeitamente o sinal LD utilizado durante o teste. Já as Figuras 4.53 e 4.54 mostram os fluxos SD e HD, respectivamente, sendo decodificados pelo receptor fixo da *Zinwell*, que também está recebendo o sinal via antena, comprovando a correta multiplexação e transmissão destes fluxos.

4.6 Conclusão

Neste capítulo foi proposto um modelo de implementação para o multiplexador, o que não é oferecido pelas normas do sistema. Facilidades propostas no modelo como o acesso via WEB aos parâmetros do multiplexador e a extração dos parâmetros do TS de forma automática se mostram bastante útil e possibilitam um fácil operação do sistema, além de ser um diferencial dos equipamentos disponíveis

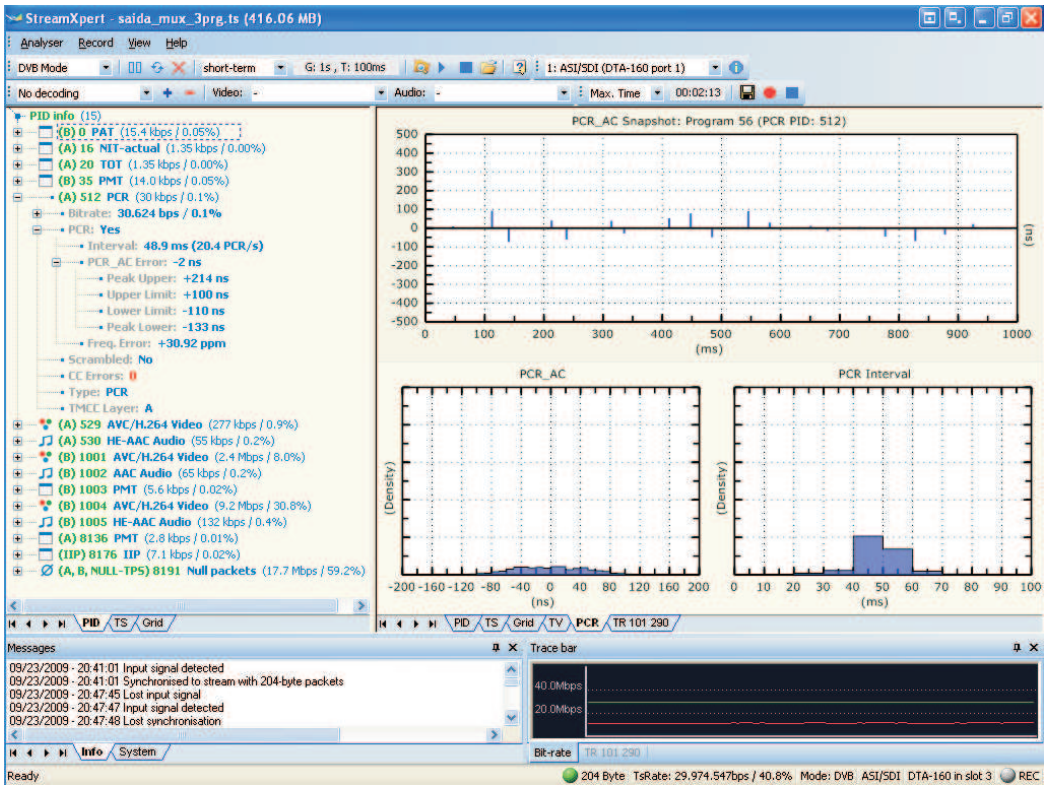


Figura 4.45: *Análise de PCR do fluxo LD do BTS.*

no mercado. Foram propostas tecnologias para a implementação deste modelo e, através de testes de validação, foi verificado que tanto o modelo quanto as tecnologias propostas são compatíveis com a necessidade do multiplexador.

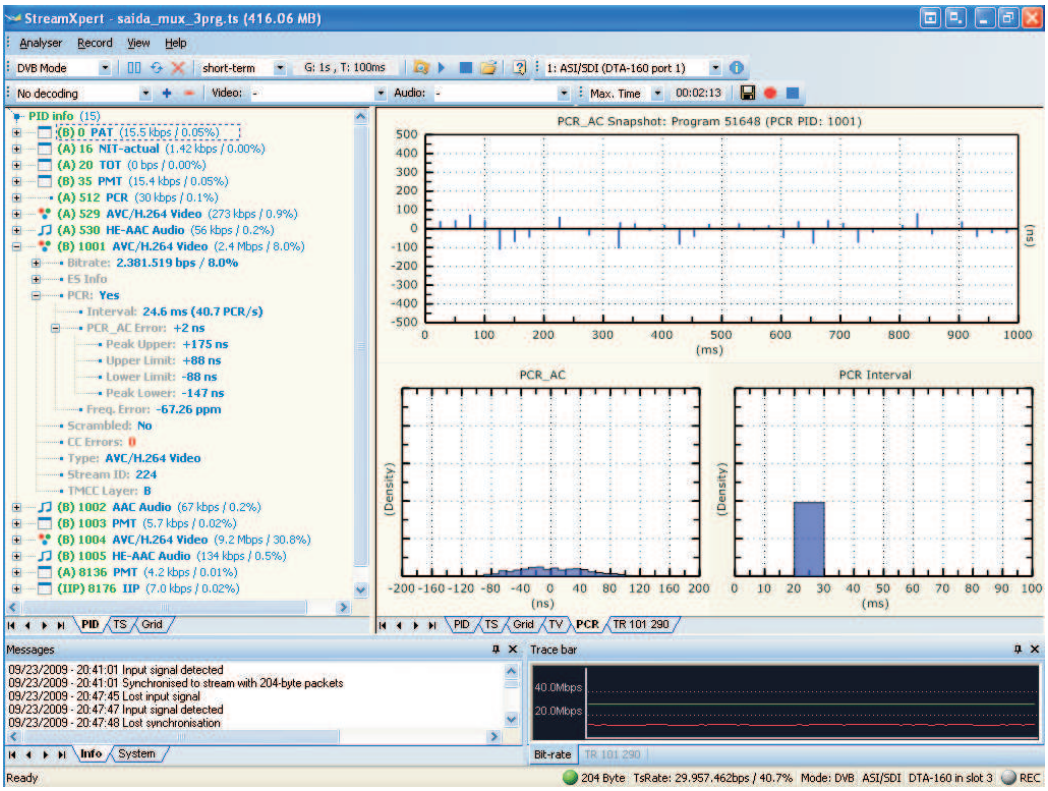


Figura 4.46: *Análise de PCR do fluxo SD do BTS.*

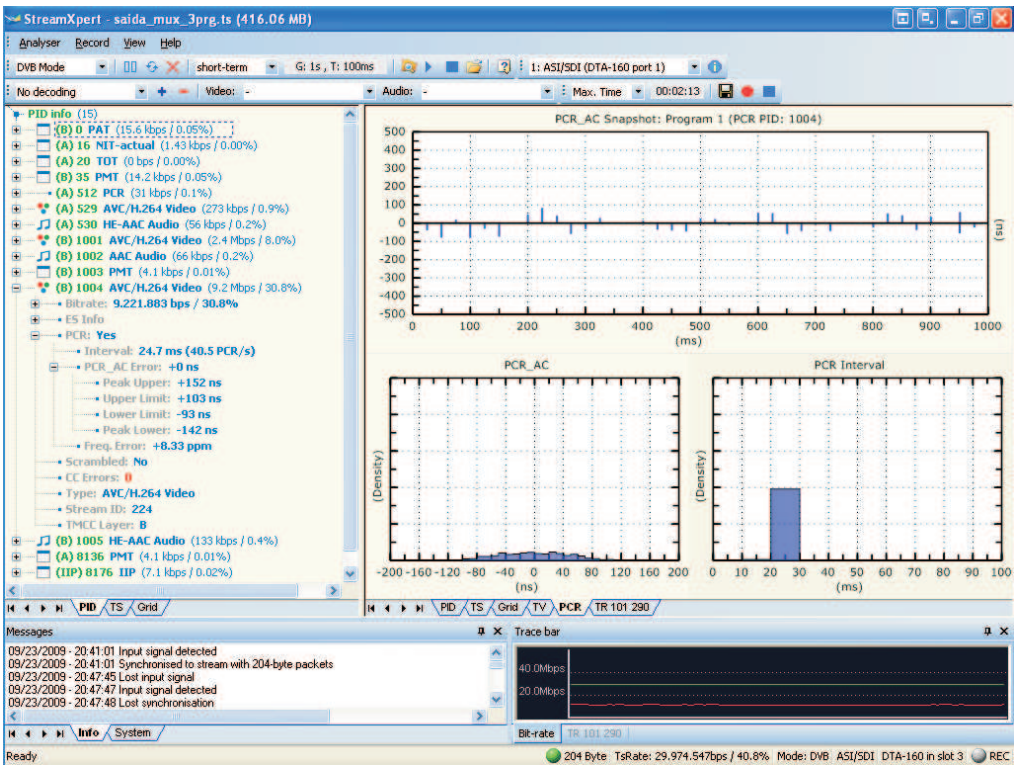


Figura 4.47: *Análise de PCR do fluxo HD do BTS.*

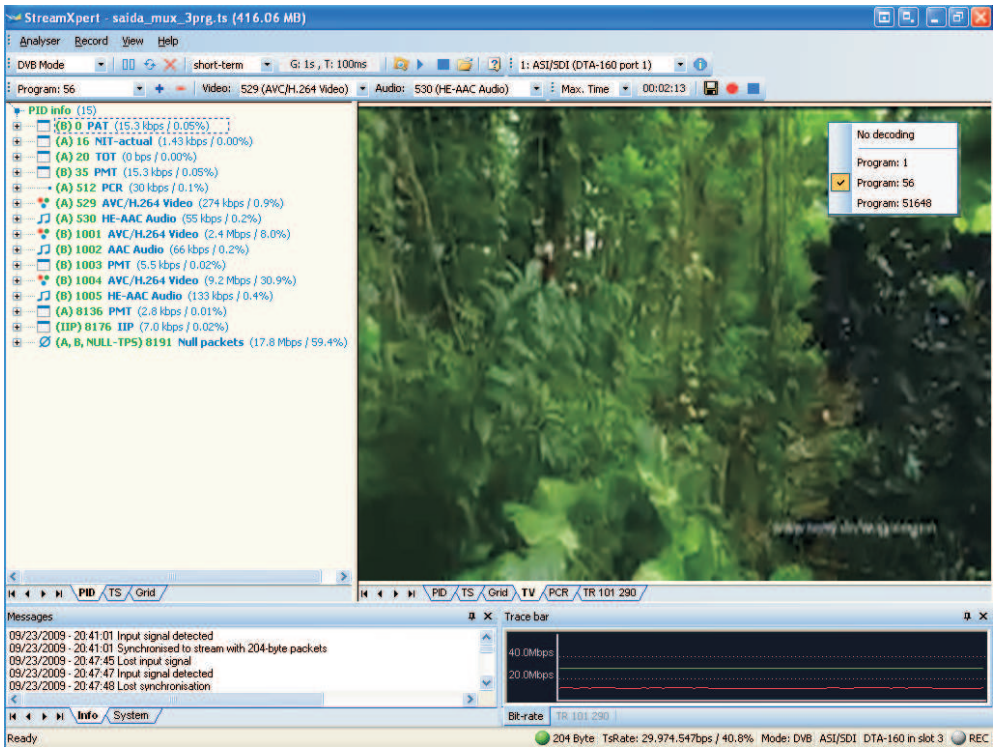


Figura 4.48: *Decodificação do fluxo LD pelo StremXpert.*

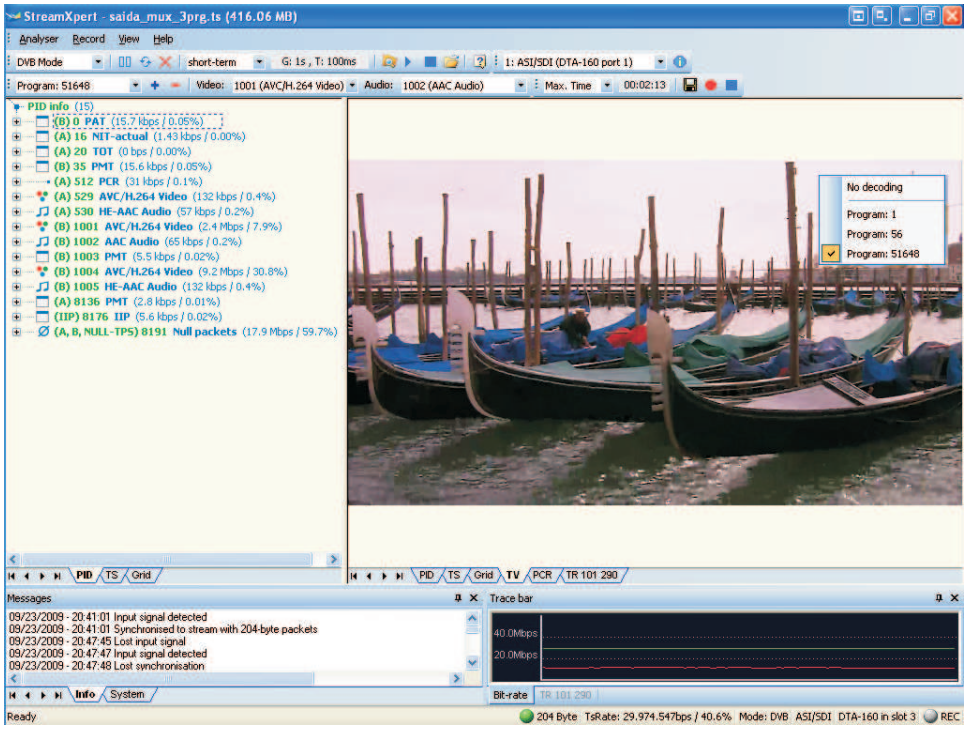


Figura 4.49: *Decodificação do fluxo SD pelo StremXpert.*

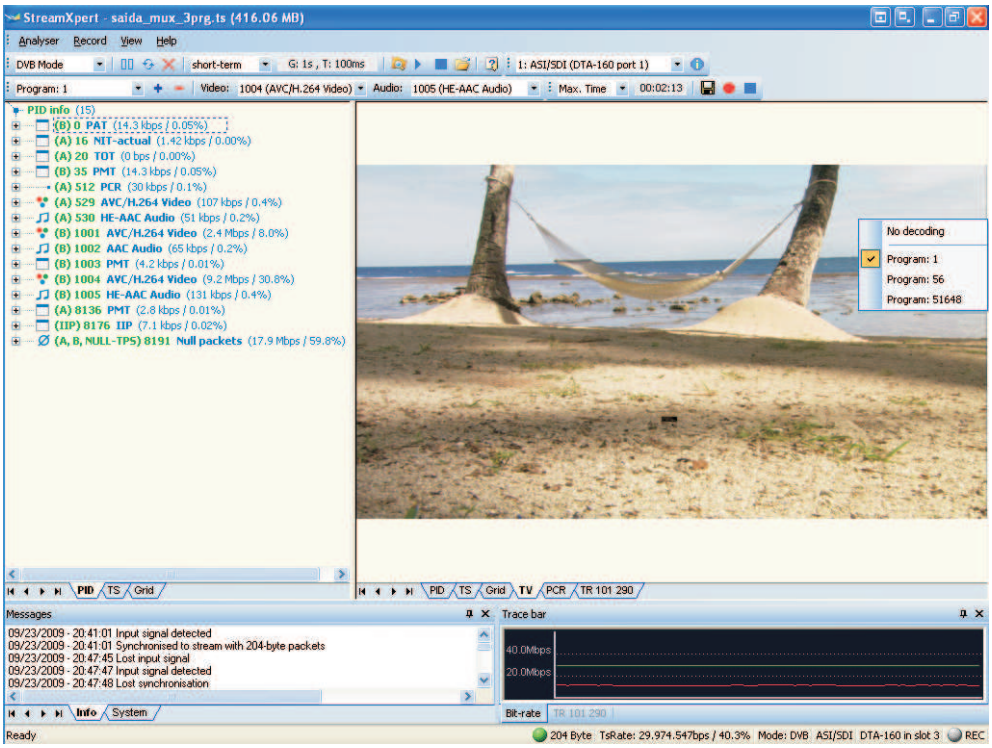


Figura 4.50: *Decodificação do fluxo HD pelo StremXpert.*

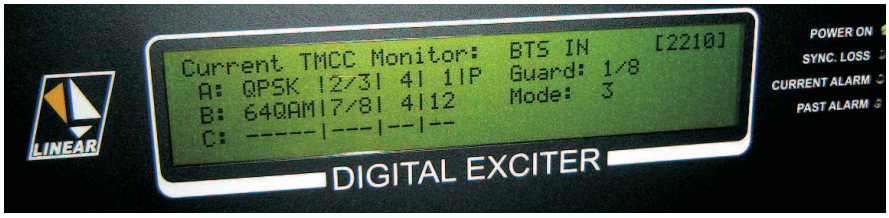


Figura 4.51: *Os parâmetros de transmissão do modulador.*

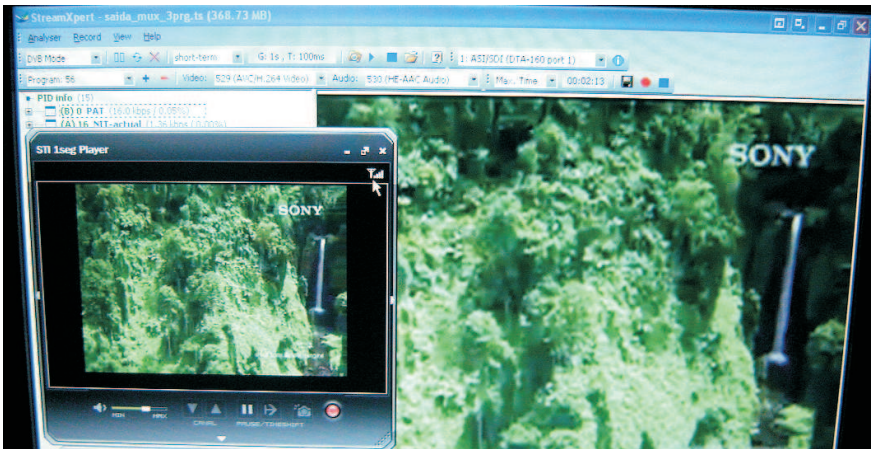


Figura 4.52: *Decodificação do sinal LD no receptor móvel e no StreamXpert.*



Figura 4.53: *Decodificação do sinal SD no receptor fixo.*



Figura 4.54: *Decodificação do sinal HD no receptor fixo.*

Capítulo 5

Conclusão

O multiplexador funciona como uma interface entre o estúdio e o transmissor da emissora de TV. Controla não só o que a emissora transmite, mas também como ela transmite. Por esses motivos, o multiplexador possui grande importância no sistema de TV digital. Sua utilização é imprescindível na grande maioria das emissoras. Portanto, sua compreensão é essencial para aqueles que atuam ou desejam atuar na área de TV digital.

Neste trabalho foram apresentados conceitos e informações necessárias para a compreensão do sistema de transmissão e multiplexação de TV digital. Essas informações sintetizadas associadas às referências bibliográficas se tornam uma referência para aqueles que iniciam sua pesquisa nessas áreas.

Neste trabalho foi proposto um modelo para a implementação do multiplexador do sistema ISDB-T_B, tendo como objetivo facilitar a operação e atualização do sistema, visto que o mesmo exige um grande conhecimento do sistema. As propostas do modelo como o acesso via WEB aos parâmetros do multiplexador e a extração dos parâmetros do TS de forma automática são as características principais do modelo proposto.

Foi mostrada uma solução de implementação do modelo proposto do multiplexador, bem como as tecnologias escolhidas para tal. Através de testes de validação, foi verificado que tanto o modelo quanto as tecnologias propostas são compatíveis com a necessidade do multiplexador. Portanto, o trabalho serve também como uma referência inicial para os pesquisadores que desejem implementar uma solução na área de multiplexação.

Como proposta a trabalhos futuros pode-se indicar a implementação de todas as tabelas e descritores estabelecidos para o sistema. Uma interessante implementação se daria com relação a geração da grade de programação da emissora, através da EIT. Outra indicação seria uma implementação do gerador de carrossel para possibilitar o encapsulamento em TS das aplicações interativas e a imple-

mentação da SDTT para a realização de atualização de *softwares* em receptores.

Anexo A

Artigos publicados desta dissertação

Os seguintes artigos baseados no conteúdo desta dissertação foram publicados:

- J. S. Ferreira; L. L. Mendes; G. G. R. Gomes, "Técnicas para Multiplexação de Sinais de TV Digital no Padrão ISDB-T_B", Simpósio Brasileiro de Telecomunicações - SBrT, Blumenau/SC - Brasil, 2009.
- J. S. Ferreira; M. C. Paiva; L. L. Mendes, "Solução de Implementação do Multiplexador do Sistema ISDB-T_B", Simpósio Brasileiro de Telecomunicações - SBrT, Rio de Janeiro/RJ - Brasil, 2008.

Referências Bibliográficas

[1] ABNT NBR 15601, "Televisão Digital Terrestre - Sistema de transmissão", primeira edição Novembro 2007.

[2] ARIB Standard STD-B31, "Transmission system for digital terrestrial television broadcasting", ver.1.6, 2007.

[3] M. Uehara, M. Takada and T. Kuroda, "Transmission scheme for the terrestrial ISDB system", IEEE *Transactions on Consumer Electronics*, Vol 45, Número 1, 1999.

[4] C. Montez V. Becker, *TV Digital Interativa - conceitos, desafios e perspectivas para o Brasil*, Editora da UFSC, 2005.

[5] L. L. Mendes e S. A. Fasolo, "Introdução à Televisão Digital", Semanana Internacional de Telecomunicações, Inatel, 2002.

[6] ABNT NBR 15603-1, "Televisão Digital Terrestre - Multiplexação e serviços de informação (SI)- Parte 1: Serviços de informação do sistema de radiodifusão", primeira edição, Dezembro 2.007.

[7] ABNT NBR 15603-2, "Televisão Digital Terrestre - Multiplexação e serviços de informação (SI) Sintaxes e definições da informação básica de SI", primeira edição, Dezembro 2.007.

[8] ABNT NBR 15602-3, "Televisão Digital Terrestre - Codificação de vídeo, áudio e multiplexação - Parte 3: Sistemas de multiplexação de sinais", primeira edição, Dezembro 2.007.

[9] ARIB Standard TR-B14, "Operational Guidelines for Digital Terrestrial Television Broadcasting", ver. 2.8, Maio 2006.

[10] ISO/IEC 13818-1, "Generic Coding of Moving Pictures and Associated Audio Information: Systems", 1998.

[11] B. Sklar, *Digital Communication - Fundamentals and Applications*, Prentice Hall, 2000.

[12] S. Haykin, *Communication Systems*, John Wiley, 2000.

[13] F. C. C. Castro, M. C. F. Castro, D. S.Arantes e Y. Iano, "Enfocando a Codificação de canal em transmissão 8-VSB para ATSC", *Telecomunicações*, Vol. 03, 2000.

[14] M. Takata and M saito, "Transmission System for ISDB-T", *Proceedings of the* IEE, Vol 94, 2006.

[15] G. Benedicks, "Results of the ISDB-T systems Tests, as Part of Digital TV study carried Out in Brazil", IEEE *Transactions on Broadcasting*, Vol. 52, 2006.

[16] W. Fischer, *Digital Television - A Practical Guide for Engineers*, Springer, 2004.

[17] MPEG-2: J. A. Nalon e Y Iano, "MPEG-2: Camada de sistemas", *Telecomunicações*, Vol. 3, 2000.

[18] *Transport Stream Compliance Analyzer*. Disponível em: <http://www2.tek.com/cmswpt/psdetails.lotr?ct=PS&cs=psu&ci=13264&lc=EN> Acessado em Setembro de 2009.

[19] MATLAB. Disponível em: <http://www.mathworks.com> Acessado em Setembro de 2009.

[20] R. Amore, "*VHDL Descrição e Síntese de Circuitos Digitais*", LTC Editora, 2005.

[21] S. Yalamanchili, "*VHDL Starter's Guide*", Prentice Hall, 1998.

[22] Quartus II *Handbook Version* 9.0. Disponível em: <http://www.altera.com/literature/lit-qts.jsp> - Acessado em Setembro de 2009.

[23] Quartus II *Handbook Version* 9.0 Volume 4: SOPC Builder. Disponível em: <http://www.altera.com/literature/lit-sop.jsp> - Acessado em Setembro de 2009.

[24] Altera *Corporation*. Disponível em: <http://www.altera.com> - Acessado em Setembro de 2009.

[25] J. S. Ferreira; M. C. Paiva; L. L. Mendes, "Solução de Implementação do Multiplexador do Sistema ISDB-TB", *SBRT*, 2008.

[26] Boa *Webserver*. Disponível em: <http://www.boa.org> - Acessado em Setembro de 2009.

[27] μcLinux - *Embedded Linux Microcontroller Projec*t. Disponível em: <http://www.uclinux.org> - Acessado em Setembro de 2009.

[28] Nios II *Processor Reference Handbook*. Disponível em: <http://www.altera.com/literature/lit-nio2.jsp> - Acessado em Setembro de 2009.

[29] Nios II *Development Kit, Stratix II Edition*. Disponível em: <http://www.altera.com/products/devkits/altera/kitverb-"niosiiverb-"2S60.html> - Acessado em Setembro de 2009.

[30] European Standard EN 50083-9, "Interfaces forCATV/SMATV headends and similar professional equipment for DVB/MPEG-2 transport streams", 1998.

[31] Linear Equipamentos Eletrônicos S.A. Disponível em: <http://www.linear.com.br> - Acessado em Setembro de 2009.

[32] Stratix II Device Family Data Sheet. Disponível em: <http://www.altera.com/literature/hb/stx2/stx2_sii5v1_01.pdf> - Acessado em Setembro de 2009.

[33] C. Costa, *Projetando controladores digitais com FPGA*, Novatec, 2006.

[34] R. Eskinazi, M. E.Lima, P. S. Nascimento e A. Guilhermino, "FPGAS dinamicamente reconfiguráveis: fluxo de projeto e vantagens na concepção de circuitos integrados", *Congresso Brasileiro de Tecnologia - CONBRATEC*, 2005.

[35] J. M. Corrêa, "Arquiteturas em FPGA para comparação de sequências biológicas em espaço linear", Tese de Doutorado em Engenharia Elétrica, *Universidade de Brasília*, 2008.

[36] M. Arnold, *Verilog Digital Computer Design: Algorithms Into Hardware*, Prentice Hall, 1998.

[37] S. Bailey, "Comparison of VHDL, Verilog and SystemVerilog", *Digital Simulation White Paper*, Model Technology.

[38] W. H. Glauert, "VHDL *Tutorial*", <http://www.vhdl-online.de/tutorial/> - Acessado em Setembro de 2009.

[39] IEEE Standard 1149.1-1990, "Standard Test Access Port and Boundary-Scan Architecture", IEEE, 1990.

[40] K. Nikkanen, "uClinux as an embedde solution", Bachelor's Thesis, Turku Polytechnic, 2003.

[41] H.S. Choi and H. C. Yun, "Context Switching and IPC Performance Comparison between uClinux and Linux on the ARM9 based Processor", *SAMSUNG Tech. Conference*, 2004.

[42] DekTec. Disponível em: <http://www.dectek.com> - Acessado em Setembro de 2009.

[43] DTC-300 StreamXpress - *Playout Software*. Disponível em: <http://www.dektec.com/Products/Apps/DTC-300/index.asp> - Acessado em Setembro de 2009.

[44] DTC-320 StreamXpert - Transport-Stream Analyser Software. Disponível em: <http://www.dektec.com/Products/Apps/DTC-320/index.asp> - Acessado em Setembro de 2009.

[45] W. Xingdong, "Implementation of MPEG-2 transport stream remultiplexer for DTV broadcasting", IEE *Transactions on Consumer Electronics*, Vol. 48, 2002.

[46] DTU-245 - *FantASI USB-2 ASI / SD-SDI*. Disponível em: <http://www.dektec.com/Products/USB2/DTU-245/index.asp> - Acessado em Setembro de 2009.

[47] ETSI ETR 290 - "Measurement guidelines for DVB systems", Maio de 1997.

[48] Doremi Labs Inc, V 1 - H D Video Servers, Manual, versão 5.

[49] Sencore, ATSC Receiver/Decoder IRD 3384A, Manual.

[50] Ateme, Ateme Kyrion 2101 Encoder, Manual.

[51] Linear Equipamentos Eletrônicos S.A., ISCHIO IS7001 - Excitador Digital UHF para ISDB-TB, Manual.

[52] onTimeTek inc, OTT 7700 Professional DMB Encoder, Manual.

[53] Zinwell, ZBT-620, Manual do Usuário para o Receptor Digital ZBT-620 Terrestre de Alta Definição.

[54] SEMP THOSHIBA, Plug TV STI / DT1020. Disponível em: <http://www.semptoshiba.com.br/produtos/mobilidade/mobilidade.aspx> Acessado em Setembro de 2009.