

Solução de implementação do multiplexador do sistema ISDB-TB

Juliano Silveira Ferreira, Marcelo Carneiro de Paiva, Luciano Leonel Mendes

Resumo—A adoção do padrão ISDB-TB pelo Brasil abriu novas possibilidades para o desenvolvimento da indústria brasileira. As ligeiras, mas importantes, modificações apresentadas pelo ISDB-TB com relação ao ISDB-T fez com que as soluções desenvolvidas para o Japão nem sempre sejam plenamente adequadas para o mercado brasileiro. Um ponto chave do sistema de TV Digital que se enquadra neste cenário é o multiplexador.

O objetivo deste artigo é, primeiramente, apresentar as principais funções do multiplexador, ressaltando sua importância para o sistema de TV Digital. Em seguida apresenta uma descrição de cada bloco do multiplexador, visando a implementação do dispositivo em uma plataforma FPGA.

Palavras-Chave— Multiplexador, ISDB-TB, implementação.

Abstract—The adoption of ISDB-TB in Brazil has opened new opportunities for the brazilian industry development. The small, but important, modifications introduced in ISDB-TB version related with the original ISDB-T standard result that the solutions developed for japanese market is not always suitable for Brazil. A key part of the digital television system that fills in this scenario is the multiplexer.

The aim of this paper is, firstly, present the main functions of the multiplexer, showing its importance in the digital television system. This paper also presents a description of each block of the multiplexer, in order to allow the implementation of this device using a FPGA platform.

Keywords— Multiplexer, ISDB-TB, implementation.

I. INTRODUÇÃO

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, com anunciantes e com a emissora. A recepção do sinal de TV poderá se dar através de diferentes tipos de receptores: fixos, móveis e portáteis.

Estas 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 ou, simplesmente, MUX do sistema ISDB (Integrated Services Digital Broadcasting) [1][2].

Este artigo irá apresentar a importância do MUX dentro do sistema e mostra uma solução para sua implementação utilizando a tecnologia FPGA (Field-Programmable Gate Array). Este trabalho está estruturado da seguinte forma: A sessão II apresenta o MUX e sua integração ao sistema de transmissão, enquanto a sessão III mostra o modelo de implementação utilizado; a sessão IV descreve cada etapa implementada e na sessão V temos a conclusão.

Juliano Silveira Ferreira, Marcelo Carneiro de Paiva, Luciano Leonel Mendes, Inatel Competence Center, Inatel, Santa Rita do Sapucaí, MG, Brasil, E-mails: silveira@inatel.br, mcpaiva@inatel.br, lucianol@inatel.br.

II. O MUX E O SISTEMA DE TRANSMISSÃO

O MUX possui um importante papel dentro do sistema de TV Digital, pois ele é o responsável por controlar não só todo o fluxo a ser transmitido pela emissora, mas também por configurar os parâmetros de transmissão.

O MUX recebe em sua entrada os vários fluxos (cada fluxo é também chamado de TS - *Transport Stream*), que a emissora deseja transmitir, além dos parâmetros de configuração do transmissor e dados adicionais. Em sua saída o MUX entrega um único fluxo denominado de BTS (*Broadcast Transport Stream*) [3]. O BTS é composto pelas informações de áudio e vídeo que a emissora deseja transmitir, além de dados utilizados para gerar o guia de programação, ajustar data e hora do receptor e permitir a interatividade.

Na Figura 1 é possível visualizar um diagrama em blocos simplificado de um sistema de transmissão utilizando o MUX.

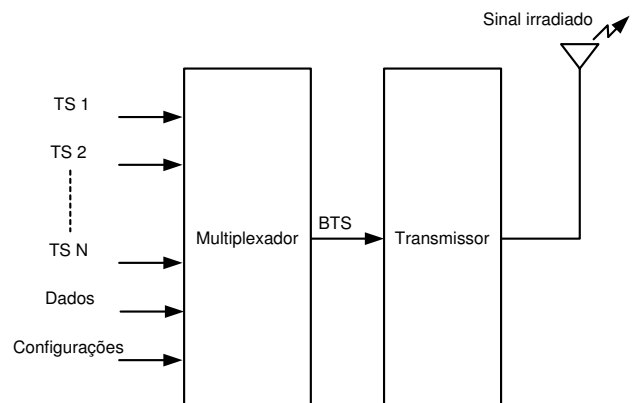


Fig. 1. O Sistema de transmissão utilizando o Multiplexador.

Cada TS corresponde a um fluxo de dados proveniente da camada de transporte do padrão MPEG-2 (*Moving Picture Experts Group*) [4], formado por pacotes de 188 bytes. Estes pacotes transportam os sinais de áudio e vídeo comprimidos entregues pelos seus respectivos codificadores ao MUX [5].

Dentre os pacotes que formam o fluxo, alguns transportam áudio, outros vídeo e outros, dados. Cada um dos pacotes possui uma identificação individual que é o chamado PID (Packet Identifier).

Para identificar o conteúdo de cada pacote no TS, a camada de transporte do MPEG utiliza um conjunto de tabelas. A PAT (*Program Association Table*) é uma tabela que lista todos os programas existentes no TS e lista o PID de cada tabela PMT, que é uma tabela que contém todos os PIDs de vídeo, áudio e dados associados a um programa específico. A Figura 2 mostra

a relação entre as tabelas PAT e PMT. Dentre as informações de sincronismo enviadas pelo codificador, o PCR (*Program Clock Reference*) merece destaque. O PCR é uma marcação de tempo inserida no codificador que permite sincronizar o clock de operação do decodificador de vídeo com clock do codificador, possibilitando a correta decodificação e exibição do vídeo.

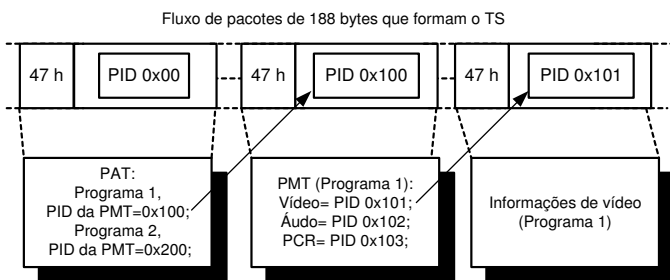


Fig. 2. Estrutura de um TS definida pelas tabelas PAT e PMT.

Na formação do BTS, algumas tabelas adicionais devem ser criadas ou atualizadas pelo MUX [6]. Por exemplo, a EIT (*Event Information Table*) é a tabela que transporta os dados referentes a grade de programação da emissora. Através da EIT é que o receptor, ou *set-top-box*, poderá montar e exibir o guia de programação ao usuário. A TOT (*Time Offset Table*) é responsável por transportar o horário e dia atual para que o *set-top-box* possa mostrar adequadamente o guia de programação e indicar corretamente qual programa está sendo apresentado a cada instante. A NIT (*Network Information Table*) é outra tabela importante que deve ser atualizada ou gerada pelo MUX. Esta tabela transporta informações a respeito dos programas e das configurações do sistema.

O BTS é um fluxo de dados que possui uma taxa fixa que é de 32,508 Mbps. É formado por pacotes de 204 bytes resultantes da adição de 16 bytes a cada pacote do TS como pode ser visualizado na Figura 3. Destes 16 bytes, 8 são de informações do sistema, como por exemplo indicação da camada hierárquica que o pacote deve ser transmitido, e os demais são bytes de paridade opcionais para correção de erros [2].

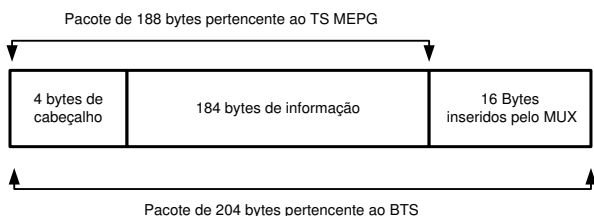


Fig. 3. Formação dos pacotes do BTS.

Na formação do BTS, deve ser acrescido ao fluxo um pacote especial que transporta todas as configurações do transmissor: número de portadoras, tempo de guarda e específica, para cada camada, o número de segmentos, taxa de código, modulação e profundidade do entrelaçador temporal. Portanto, este pacote, denominado de IIP (ISDB-T Information Packet) [2], transporta dados que são utilizados pelo transmissor para sua

configuração e serão conseqüentemente transmitidos nas portadoras TMCC (*Transmission and Multiplexing Configuration Control*) [1][2].

III. MODELO PARA IMPLEMENTAÇÃO DO SISTEMA DE MULTIPLEXAÇÃO

A implementação do MUX foi realizada utilizando-se FPGA. A escolha por FPGA se deve ao fato desta tecnologia oferecer uma plataforma de processamento dedicada ao projeto desenvolvido, diferente por exemplo do desenvolvimento de uma solução em software, que teria de dividir o poder de processamento da CPU entre as requisições do sistema operacional, dos periféricos e do próprio software do projeto. Outro fator determinante na escolha pelo FPGA foi a necessidade de controle rigoroso de taxas e clocks. Por exemplo, o sinal de entrada e saída do MUX se dá através de uma interface que possui uma taxa fixa de 270 Mbps denominada de ASI (*Asynchronous Serial Interface*). A taxa de bits úteis gerada na saída do MUX deve ser fixa em 32,508 Mbps. Através de um sistema dedicado, como o FPGA, fica mais simples garantir a estabilidade de geração destes sinais nas taxas especificadas.

O modelo proposto para implementação do MUX pode ser visualizado na Figura 4.

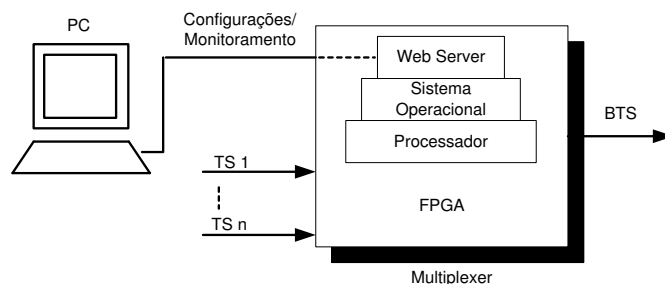


Fig. 4. Diagrama do modelo do MUX implementado.

A principal linguagem de programação utilizada para implementação do projeto foi a linguagem VHDL (*VHSIC Hardware Description Language*) [7] associada a códigos escritos em linguagem C. As ferramentas de desenvolvimento utilizadas foram o *Quartus II* e *SOPC Builder*, ambos da Altera [9].

O FPGA da empresa Altera, Stratix II modelo EP2S60F672 com 60.440 elementos lógicos e 2.544.192 bits de memória RAM [10] foi empregado para a implementação do protótipo.

A interface com o usuário, para configuração do MUX, foi construída utilizando linguagem HTML, que foi suportada por um servidor Web. O servidor Web é executado sobre um sistema operacional residente em uma CPU (*Central Processing Unit*) dentro do FPGA, denominada de Nios II.

O Nios II é um processador RISC de propósito geral que foi desenvolvido pela Altera para ser embarcado dentro de seus FPGA's. Este é um componente versátil, que pode ser configurado de acordo com a necessidade do projeto e pode-se utilizar a interface JTAG (*Joint Test Action Group*) para gravar e testar programas no dispositivo.

Conforme comentado na sessão II, o MUX deve inserir no seu fluxo de saída novas tabelas compatíveis com o TS MPEG-2 e com sistema ISDB-TB.

Os principais multiplexadores encontrados no mercado desenvolvidos em hardware necessitam de um software externo para geração destas tabelas. Neste software o usuário normalmente deve cadastrar manualmente todas as informações necessárias para o correto funcionamento do sistema.

Na implementação em questão, a geração das tabelas é realizada de maneira automática pelo MUX. Ele se encarrega de analisar os fluxos de entrada, extrair as informações de configuração do sistema de transmissão e gerar as tabelas necessárias. Um grande vantagem desta 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 destes parâmetros, novas tabelas são geradas levando em consideração as informações atuais.

A geração destas 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. Estes códigos são executados pela CPU embarcada no FPGA.

IV. IMPLEMENTAÇÃO E INTEGRAÇÃO DO SISTEMA DE MULTIPLEXAÇÃO

Um diagrama resumido do projeto implementado pode ser visualizado na Figura 5.

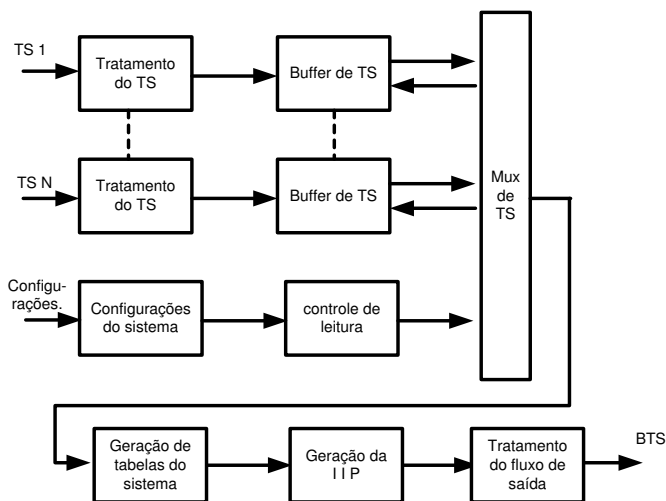


Fig. 5. Diagrama geral do MUX implementado.

Cada um dos blocos do sistema estão descritos nas sessões que seguem.

A. Tratamento do TS de entrada

Esta etapa é responsável por receber o TS através da interface ASI e retirar as informações necessárias para o sistema gerar novas tabelas de transmissão. Esta etapa também

realiza parte do tratamento de PCR. O sinal entregue por esta etapa é encaminhado ao buffer de TS. A Figura 6 mostra as etapas que compõem o tratamento do TS de entrada, descritas sucintamente a seguir.

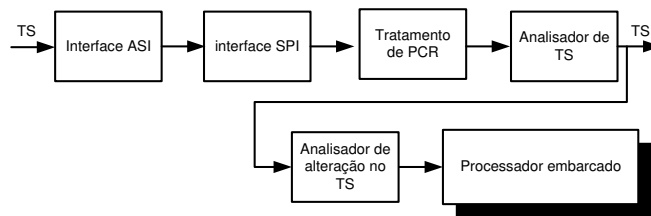


Fig. 6. Tratamento do TS de entrada.

1) *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 [11]. A implementação desta etapa possui um ponto crítico que é o alto valor do clock de referência dos dados recebidos. Deve-se ter um cuidado especial para se atender os requisitos de tempo e garantir a correta amostragem dos dados. Na solução implementada, fixou-se a localização dos registros que recebem os dados de entrada para que estes ficassem o mais próximo do pino físico de entrada no FPGA, reduzindo o tempo de propagação dentro da pastilha.

Os dados recebidos serialmente são amostrados na transição de subida e de descida por dois clock locais: um de 202 MHz e outro de mesmo valor, porém, defasado de 90°. Em média, são tomadas 5 amostras por bit, totalizando uma taxa de amostragem dos dados recebidos de 1.350 Mbps.

As amostras dos dados são re-amostradas usando o mesmo sinal clock e convertidas em palavras de 10 bits. Os pontos de transição dos dados são determinados e seleciona-se, dentre as amostras, aquela com informação mais estável.

As amostras selecionadas são repassadas, tendo como referência um clock de 135 MHz, ao decodificador 8B10B que irá detectar e corrigir erros nos bits recebidos, entregando em sua saída 8 bits. Caracteres especiais enviados para garantir a taxa fixa de 270 Mbps são eliminados e os dados (8 bits em paralelo) são repassados às demais etapas, tendo como referência um clock de 27 MHz.

2) *Interface SPI*: esta etapa é responsável por detectar o sincronismo dos pacotes do TS e entregar em sua saída os dados em uma interface SPI (*Synchronous Parallel Interface*), 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.

A busca pelo sincronismo do TS é iniciada a partir do momento em que é localizado o primeiro byte 47h (valor do byte de sincronismo) [4]. A partir deste ponto, é verificado se no 188º byte seguinte possui este mesmo valor; se for confirmado, o teste prossegue para o próximo pacote. Caso haja 3 confirmações em sequência, admite-se que o sincronismo de início de pacote foi localizado. Nesse instante, os dados passam a ser entregues no formato de SPI 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. Esta etapa sinaliza ainda a presença ou ausência de TS em sua entrada.

3) *Tratamento do PCR*: há necessidade de se realizar atualização do valor do PCR, devido ao fato de se realizar alterações no fluxo de dados, como armazenamento em buffer, retirada e inserção de pacotes nulos e multiplexação de pacotes de diferentes TS, o que altera o espaçamento original dos pacotes do TS.

O PCR é a referência do valor de dois contadores: um de 33 bits e outro de 9 bits que possuem como clock de referência um oscilador de 27 MHz (embora um conte em unidades de 900 KHz e o outro em unidades de 27 MHz, respectivamente).

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 quando o pacote é entregue à saída [8]. Na primeira fase o valor do PCR do pacote recebido é subtraído do valor de um PCR local. Na segunda o valor do PCR do pacote é somado ao valor do PCR local. Esta técnica acrescenta 50 ns no valor do *jitter* do PCR, que por norma não deve exceder 500 ns [8]. Esta etapa implementa a primeira fase da atualização de PCR, conforme pode ser visualizado na Figura 7.

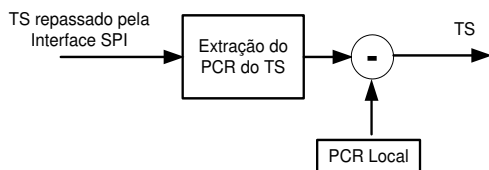


Fig. 7. Primeira fase do tratamento do PCR.

4) *Analisador de TS*: esta etapa monitora e altera o fluxo que é encaminhado ao buffer de TS, além de extrair informações necessárias para se construir novas tabelas como a PAT e NIT. Este trabalho se dá através da análise dos pacotes que transportam a PAT (pacotes com PID=0x00). As informações extraídas são a identificação (*Program Numbers*) dos programas do fluxo e PIDs das respectivas PMTs.

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

5) *Analisador de alterações no TS*: as informações extraídas pelo analisador de TS são monitoradas por esta etapa. Verifica-se constantemente se não há alterações nestas informações. Quando qualquer diferença é detectada, um pedido de interrupção é enviado ao processador embarcado, solicitando permissão para o envio dos dados atualizados.

B. O Buffer de TS

O buffer de TS foi implementado utilizando-se uma memória FIFO (*First In First Out*) de 4096 bytes, capaz de armazenar pouco mais de 21 pacotes de 188 bytes. Um controle de nível foi implementado para controlar a leitura e escrita

sobre a FIFO. Como dito anteriormente, os pacotes nulos não serão armazenados. Quando, em uma eventual leitura, for detectado que existe apenas 1 pacote de 188 bytes no buffer, um pacote nulo é gerado e entregue à saída ao invés de se realizar a leitura da FIFO. No caso da escrita, quando o espaço disponível na FIFO não for suficiente para armazenar 1 novo pacote de 188 bytes, o pacote de recebido será descartado. Os dados recebidos na entrada da FIFO possuem como referência um clock de 27 MHz. O clock de leitura tem como base um clock de 4,063 MHz.

C. Configurações do Sistema

O MUX 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.

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 MUX: Modo de operação, tempo de guarda; para cada camada hierárquica deve-se especificar o número de segmentos, modulação, taxa de código e profundidade do entreteçador 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 CGIs (*Common Gateway Interface*) implementados com a linguagem C. Na Figura 8 pode-se visualizar a página Web através da qual é possível configurar os parâmetros do transmissor.

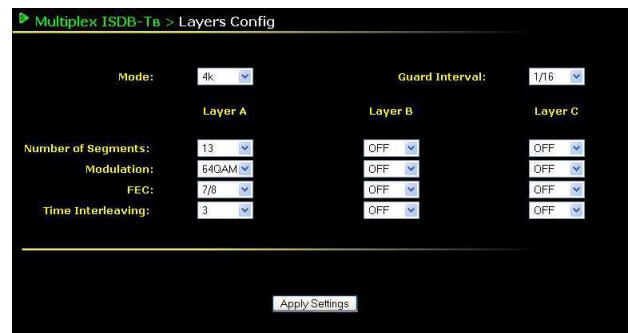


Fig. 8. Tela de configuração dos parâmetros do transmissor.

Para disponibilizar as aplicações em rede, foi utilizado o servidor Boa Webserver [12] e o sistema operacional μ CLinux [13]. Ambos são gratuitos, de código aberto e foram desenvolvidos para serem utilizados em sistemas embarcados.

D. MUX de TS

O MUX de TS é o responsável por multiplexar os vários fluxos a serem transmitidos. No instante em que recebe a solicitação de um pacote a ser transmitido na camada A, por exemplo, o MUX de TS analisa o nível de todos os buffers de TS que contem fluxos a serem transmitidos nesta camada, e realiza leitura do buffer que possuir maior número de pacotes armazenados.

E. Controle de leitura do MUX de TS

O controle de leitura dos pacotes para formação do fluxo de saída do MUX, que é o BTS, é realizada por esta etapa. Ela solicita ao MUX de TS pacotes de dados e informa, a cada solicitação, em qual camada hierárquica tal pacote será transmitido. Estas solicitações possuem uma ordem e periodicidade que são determinadas pela lei de formação do quadro de multiplexação [2][14].

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 e de 32.508 Mbps.

F. 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 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 tempo real (RTC - *Real Time Clock*). 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 atualizada.

As tabelas devem ser transmitidas no BTS respeitando um determinado intervalo de tempo [6]. A PAT é enviada a cada 100 ms em média, 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.

G. Geração da IIP

Esta 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 1 pacote IIP a cada quadro de Multiplexação [2].

Quando o transmissor identifica que está recebendo um pacote IIP, que possui um PID especial (PID 0x1ff0), extrai do pacote as informações de configuração necessárias para sua operação. Estas mesmas informações são transmitidos nas portadoras TMCC (*Transmission and Multiplexing Configuration Control*) do sinal irradiado pelo transmissor. As informações contidas nas portadoras TMCC são utilizadas pelo receptor, para que este possa se configurar de maneira idêntica ao transmissor e poder, assim, tratar corretamente o sinal recebido. Qualquer alteração nos parâmetros de transmissão realizada no MUX irá ser refletida em todo processo de transmissão e recepção do sinal.

H. Tratamento do TS de saída

Esta etapa é a responsável por tratar o TS proveniente das etapas anteriores e o entregar à saída do MUX. O tratamento realizado envolve a mudança de domínio de clock dos dados, implementação da segunda fase do Tratamento do PCR e conversão do fluxo à um sinal compatível com a interface ASI. Estas tarefas são resumidamente tratadas a seguir.

1) *Mudança de domínio de clock*: os dados repassados pelas etapas anteriores tem como referência um clock de 4,063492 MHz. Estes 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.

2) *Tratamento do PCR*: implementa a segunda fase do tratamento do PCR apresentado na sessão IV. Na Figura 9 está mostrada a operação realizada nesta etapa.

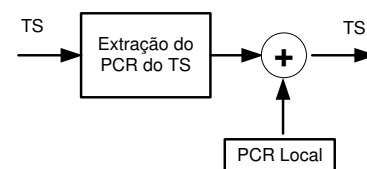


Fig. 9. Segunda fase do tratamento do PCR.

3) *Interface ASI de saída*: realiza a codificação 8B/10B sobre os bytes recebidos, convertendo-os em 10 bits. Estes 10 bits são enviados serialmente à saída do MUX, 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.

A Figura 10 apresenta uma foto do protótipo implementado, onde foi utilizado como base um *kit* de desenvolvimento da Altera (Nios II Development Kit - EP2S60) associado a uma placa de 4 entradas e 2 saídas ASI desenvolvida pela empresa Linear Equipamentos Eletrônicos S/A.

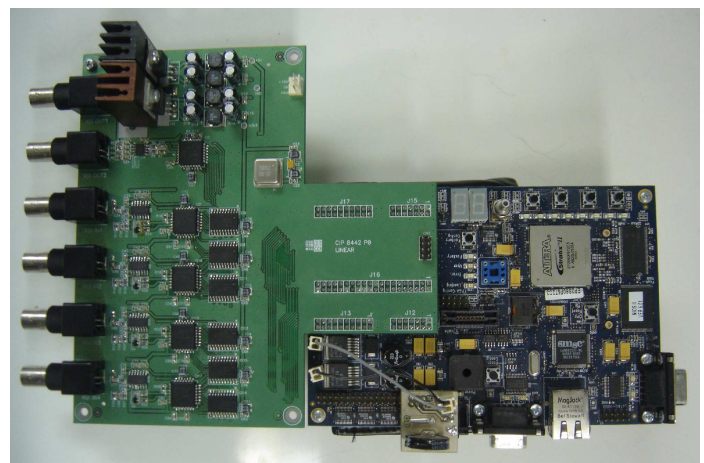


Fig. 10. Foto do protótipo do Multiplexador.

V. CONCLUSÕES

O sistema implementado apresentou estabilidade e funcionamento compatível com transmissores e receptores disponíveis no mercado. Testes foram realizados com transmissores das empresas Eiden e Linear e com receptores das seguintes empresas: *set-top-box* Zinwell e Aiko, celular Samsung, receptor para automóveis da Gradiente e receptor USB da Sanwa.

Através do software de análise de fluxos, o StreamXpert desenvolvido pela empresa Dectek [15], pode-se validar a compatibilidade do fluxo de saída do MUX, o BTS, com o especificado pelo sistema ISDB.

Através de testes práticos, integrando MUX e transmissor ISDB-TB (tanto das empresas EIDEN quanto da LINEAR), pode-se verificar o correto funcionamento do multiplexador na utilização de multi-programação (transmissão de mais de 1 TS de entrada na camada B) e transmissão simultânea para receptores fixos e móveis (transmissão hierárquica: camada A e B).

A solução implementada mostrou atender as necessidades reais do mercado de TV digital atual, além de ser uma solução de baixo custo e de fácil utilização.

AGRADECIMENTOS

Agradecimentos ao INATEL - Instituto Nacional de Telecomunicações - e à Linear Equipamentos Eletrônicos S/A por oferecerem a estrutura e o suporte necessário para o desenvolvimento deste trabalho. Agradecimento especial aos engenheiros Bruno Borsato de Souza Lima, Cleomárcio Almeida, Dário Daniel Ribeiro de Moraes, Farley Igor Martins Balbino e Rafael Adami Pivoto, que contribuíram de maneira expressiva para esta implementação.

REFERÊNCIAS

- [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] Yasuo Takahashi, *Seminário técnico ISDB*, SET, 2007.
- [4] ISO/IEC 13818-1, *Generic coding of moving pictures and associated audio information: Systems*, 2000.
- [5] José A. Nalon e Yuzo Iano *MPEG-2: Camada de sistemas*, Revista Científica Periódica - Telecomunicações, Volume 03, Abril de 2000.
- [6] ABNT NBR 15603, *Televisão digital terrestre - Multiplexação e serviços de informação (SI)*, primeira edição, Novembro 2007.
- [7] Sudhakar Yalamanchili, *VHDL Starter's Guide*, Prentice Hall, 1998.
- [8] Wang Xingdong, *Implementation of mpeg-2 transport stream remultiplexer for DTV broadcasting*. IEEE, 2002.
- [9] Altera Corporation. Disponível em: <<http://www.altera.com>> Acessado em Junho de 2008.
- [10] Stratix II Device Family Data Sheet. Disponível em: <http://www.altera.com/literature/hb/stx2/stx2_sii5v1_01.pdf> Acessado em Junho de 2008.
- [11] European Standard EN 50083-9, *Interfaces for CATV/SMATV headends and similar professional equipment for DVB/MPEG-2 transport streams*, 1998.
- [12] Boa Webserver. Disponível em: <<http://www.boa.org>> Acessado em Junho de 2008.
- [13] µcLinux - Embedded Linux Microcontroller Project. Disponível em: <<http://www.uclinux.org>> Acessado em Junho de 2008.
- [14] ARIB Standard TR-B14, *Operational guidelines for digital terrestrial television broadcasting*, ver. 2.8, Maio 2006.
- [15] DekTec. Disponível em: <<http://www.dectek.com>> Acessado em Junho de 2008.