

Metodologia de Reconfiguração de Hardware utilizando o Sinal de TV Digital

Rodrigo Ribeiro de Oliveira
Universidade Federal do Amazonas
Av. Rodrigo Octávio J. Ramos, 3000
69077000, Coroado I. Manaus, AM
+55(092)81143441

rodrigo@dcc.ufam.edu.br

Eddie Batista de Lima Filho
Centro de Ciência, Tec. e Inov. do PIM
Rua Salvador, 391 69057040,
Adrianópolis, Manaus, AM
+55(092)21235833

eddie@ctpim.org.br

Vicente Ferreira de Lucena
Universidade Federal do Amazonas
Av. Rodrigo Octávio J. Ramos, 3000
69077000, Coroado I. Manaus, AM
+55(092)81586371

vicente@ufam.edu.br

RESUMO

O presente artigo apresenta um modelo de Set-top Box (STB) com decodificador de vídeo reconfigurável, funcionando a partir do sinal aberto de TV Digital (TVD). O modelo foi baseado em uma plataforma comercial e utiliza um módulo de *Field Programmable Gate Array* (FPGA) para o processo de reconfiguração, no qual é prevista a atualização do decodificador de vídeo H.264. O sistema efetua a reconfiguração do FPGA a partir de um feixe de bits contendo a descrição de hardware (H.264), o qual é transmitido juntamente com o conteúdo de TV em alta definição (*High Definition Television* -- HDTV). Dessa maneira, todos os receptores na área de cobertura da transmissora de TVD podem ser atualizados através de um único sinal. A transmissão e a recepção dos dados de atualização fazem parte do modelo conceitual proposto, cujo desenvolvimento objetiva a minimização do legado normalmente existente na implantação ou na evolução de um sistema de TVD. Assim, futuras revisões nas normas de TVD poderiam ocorrer sem a necessidade de troca de equipamento.

Palavras-chave

HDTV, TVD, DVB-T, ISDB-T, SBTVD, STB, H.264, FPGA VHDL, Verilog, NRE e Datacasting.

1. INTRODUÇÃO

A reconfiguração de hardware já é amplamente utilizada em diversos segmentos da indústria de eletroeletrônicos e tem apresentado crescimentos substanciais na área de sistemas embarcados [11]. Estudos recentes indicam que o uso da reconfiguração de hardware pode trazer benefícios para o gerenciamento do ciclo de vida dos produtos e contribuir também com a redução dos custos de engenharia não recorrente (*Non-Recurring Engineering* -- NRE). Tradicionalmente, os circuitos integrados de aplicação específica (*Application-Specific Integrated Circuits* - ASICs) oferecem o menor custo unitário. No entanto, pressões crescentes para a colocação de produtos no mercado (*time to market*) aumentam exponencialmente os custos de NRE [3]. Uma máscara definida para um ASIC no processo de 90nm custa cerca US\$ 1,500,000.00; por outro lado, para grandes volumes de chips comercializados, o custo unitário do FPGA, que é geralmente alto, é equiparado ao custo do ASIC, o que é essencialmente dado pela amortização dos custos de NRE entre os clientes de cada chip. Essa compensação ocorre tendo-se em vista que os custos de NRE são menores no FPGA [3][15].

Os STBs, por possuírem um amplo mercado, são comercializados em larga escala (individualmente ou integrado em um televisor),

reforçando a utilização da reconfiguração de hardware no controle do ciclo de vida do produto final, que está associado diretamente à arquitetura na qual o sistema foi concebido. Atualmente, alguns STBs disponíveis no mercado utilizam decodificadores de vídeo desenvolvidos em hardware e outros adotam processadores de maior capacidade e decodificadores em *software*. Dessa forma, a preparação do sistema para o processo de atualização deve ser realizada na concepção do projeto, de acordo com a estrutura de funcionamento desejada.

Um sistema de reconfiguração de hardware faz uso de lógica programável para determinar o funcionamento do FPGA, a partir de uma descrição de hardware (neste caso, contida no feixe de bits). A síntese do feixe de bits é realizada através de uma ferramenta de projeto assistido por computador (*Computer-Aided Design* -- CAD) que, por sua vez, é baseada em um código com a descrição do hardware (gerado a partir de uma linguagem de descrição de hardware, ou *Hardware Description Language* -- HDL, como VHDL e Verilog). No contexto de um sistema de TVD, o feixe de bits pode ser visto como um dado binário que é transmitido juntamente com o conteúdo de HDTV da programação normal (Figura 1).

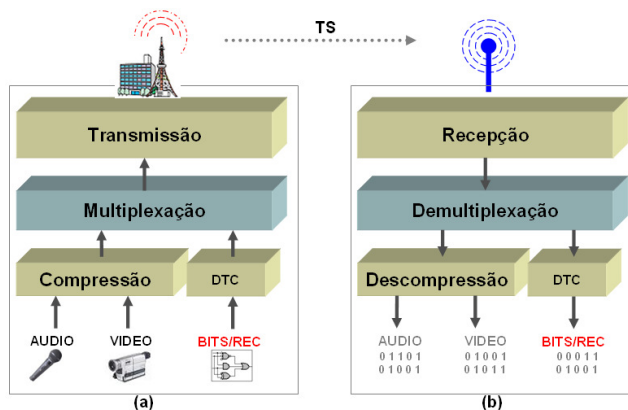


Figura 1: Transmissão do conteúdo de reconfiguração de hardware.

Um sistema de transmissão e recepção do sinal de TVD é composto por diversos subsistemas, que contemplam desde a preparação dos dados para a transmissão até a completa recepção do sinal e posterior filtragem do conteúdo enviado.

O conteúdo é composto por pacotes de áudio, vídeo e dados. Neste caso, os dados de reconfiguração de hardware são carregados através de um mecanismo de difusão de dados

(*datacasting* -- DTC). A Figura 1(a) descreve as três etapas básicas envolvidas na transmissão do sinal de TVD. Na etapa de Compressão, os dados de áudio e vídeo são codificados (comprimidos) em MPEG-4 (vídeo em H.264/AVC e áudio em HEAAC), considerando-se o Sistema Brasileiro de Televisão Digital (SBTVD). O feixe de bits com a descrição do hardware é encapsulado através de um mecanismo de DTC (ver capítulo 3). Na fase de multiplexação, áudio, vídeo e dados com a descrição de hardware são entrelaçados, formando um feixe de transporte (*Transport Stream* -- TS ou MPEG2-TS), o qual é necessário para a transmissão de informação em um sistema de TVD. Por fim, na etapa de transmissão é realizada a modulação e o sinal é então transmitido. A Figura 1(b), está ilustrada a recepção do sinal, que é primeiramente capturado e demodulado. O resultado desse processo inicial é então demultiplexado, extraindo-se pacotes de áudio, vídeo e dados, sendo que estes últimos são utilizados para a reconfiguração do FPGA.

A reconfiguração é efetuada com a interação do usuário, através do sistema de controle residente no STB, que notifica o usuário quando novas atualizações estão disponíveis. A partir do aceite do usuário, o sistema obtém o feixe de dados de reconfiguração e o grava na memória permanente do STB. Após isso, o FPGA é reiniciado automaticamente, já com uma nova descrição de funcionamento.

Este artigo está dividido em cinco seções. A seção 2 contém uma breve explanação sobre a reconfiguração de hardware e as tecnologias associadas a tal processo. A seção 3 discorre sobre a tecnologia de TV digital e os mecanismos de transmissão e recepção. Na seção 4, é apresentada a arquitetura da plataforma de reconfiguração. A seção 5 expõe as considerações finais e a seção 6 os agradecimentos.

2. RECONFIGURAÇÃO DE HARDWARE

A reconfiguração de hardware é um meio flexível para o desenvolvimento de circuitos. Os recursos da reconfiguração de hardware (*Flip-Flops*, *LUTs*¹, *Slices*² e *Registers*) são configuráveis (e re-configuráveis) após a fabricação, permitindo que, em um único dispositivo, uma grande variedade de circuitos seja desenvolvida [11]. Uma memória é utilizada para manter os dados de configuração (descrição do hardware contida no feixe de bits).

Um grande crescimento do uso de FPGAs já é observado em diversos segmentos da indústria de eletroeletrônicos [11], inclusive na área de TV Digital, para a implementação de funções específicas do receptor, como controle de temporização e interfaces de saída [7]. Entre os principais avanços da tecnologia de FPGA, destacam-se o mapeamento lógico e o fluxo de implementação, atualmente similar ao fluxo de projeto do ASIC [3][6]. Outro fator importante, que deve ser levado em consideração, é que o setor industrial disponibiliza uma acervo de códigos de referência, como circuitos e módulos em HDL, *bitstreams* (feixes de bits) sintetizados para diferentes arquiteturas e também HDLs para simulação e validação de circuitos [3][6].

Na área de TVD, foco deste trabalho, pesquisas vistas em [4] e [5] mostram o uso do FPGA no processo de decodificação de vídeo

H.264. Os resultados alcançados são satisfatórios e o uso destes dispositivos é recomendado para os receptores de TVD. Conforme [5], as pesquisas nessa área têm o objetivo de somar esforços no desenvolvimento de propriedade intelectual para o desenvolvimento do SBTVD³. Em [5], o dispositivo alvo (FPGA) utilizado para o processo de síntese foi um XILINX XC2VP30 VIRTEX-II Pro. Os recursos necessários a esse processo contemplam 21200 LUTs, 8465 slices, 5671 registradores, 21 módulos de memória interna e 12 multiplicadores. Já em [4], o dispositivo alvo utilizado no processo de validação e teste foi um XILINX 2VP30FF896 VIRTEX II - Pro. Nesse caso, os recursos necessários são apresentados na Tabela 1. A coluna mais a direita mostra o percentual de uso dos recursos e a coluna Número tem a quantidade de elementos utilizados.

Tabela 1: Consumo necessário para o processo de síntese.

	Número	Porcentagem
Slices	4246 de 13696	31%
Slice Flip Flops	5849 de 27392	21%
4 Input LUTs	3249 de 27392	11%
Bonded IOBs	109 de 556	19%
MULT18x18s	3 de 36	2%
GCLKs	1 de 16	6%

Fonte: [4].

Os resultados apresentados em [5] e [4] são tomados como referências para a escolha do dispositivo mais adequado a este projeto. A escolha do dispositivo deve considerar o tempo de vida do produto e também a quantidade de recursos que devem ficar disponíveis para as futuras atualizações.

3. TECNOLOGIA DE TV DIGITAL

Uma das principais propriedades de um sistema de TV Digital é a difusão de dados (*datacasting*). O *datacasting* permite que as transmissoras forneçam serviços de entrega de informação a usuários que utilizam receptores (*Set-top Boxes*) adequados ao recebimento da mesma, desde que estejam localizados na área de cobertura da emissora. Em um sistema de TV Digital, os conteúdos de áudio, vídeo e dados e as demais informações relacionadas ao conteúdo da programação são organizados em pacotes (Figura 2) e multiplexados em um TS [8], antes de serem enviados por equipamentos de transmissão.

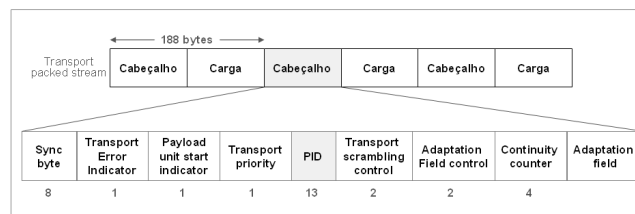


Figura 2: Estrutura do pacote de um transporte stream.

O pacote de transporte é a unidade básica de dados em um TS. Ele consiste em um byte de sincronismo (*sync byte*), cujo valor é 0x47, seguido de 3 campos de 1 bit cada (*transport error indicator*, *payload unit start indicator* e *transport priority*) e os 13 bits do PID (*Packet Identifier*). O PID é o identificador de pacote e proporciona um meio de se distinguir as várias unidades de transporte em um TS; se um PID for alocado em alguma tabela ou informado ao receptor, isso significa que um determinado

¹ Look-Up Table, que é um bloco de memória utilizado para implementar pequenas funções lógicas.

² Slice, que na família Virtex da Xilinx é a menor unidade lógica configurável.

³ SBTVD – Sistema Brasileiro de TV Digital.

pacote carrega vídeo, áudio ou outros dados. Sendo assim, nos sistemas proprietários os conteúdos difundidos são identificados através de seus respectivos valores de PID. O campo de adaptação (*adaptation field*) deve obrigatoriamente ser utilizado para transmitir informações adicionais do cabeçalho. A presença desse campo é opcional. O campo de carga (*payload*) deve obrigatoriamente ser utilizado para transmitir pacotes PES ou seções [1].

3.1 Mecanismos de Datacasting

Os mecanismos de *datacasting* permitem a inserção de dados durante o processo de multiplexação do TS. O MPEG-2 é o padrão de fato e de direito para a multiplexação de dados em sistemas de TV Digital abertos. Dessa forma, os mecanismos de *datacasting* fazem uso das facilidades e estruturas encontradas nesse padrão para efetuar a difusão de dados. São definidos quatro mecanismos de transporte: *Data Piping*, *Data Streaming*, *Multiprotocol Encapsulation* (MPE) e *Carousel*, sendo esse último dividido em *Data Carousel* e *Object Carousel* [10]. Nas próximas seções, serão apresentadas as características de cada um destes mecanismos.

3.1.1 Datapiping

Data Piping, é o mecanismo mais simples de *datacasting* e baseia-se na inserção de dados brutos diretamente na carga dos pacotes de transporte [10]. A tarefa de um encapsulador de dados desse tipo é gerar pacotes de transporte com um valor do PID conhecido pela aplicação destino, inserindo dados nos 184 bytes da carga de cada um antes de enviá-los ao multiplexador. Durante o processo de recepção dos dados, o receptor precisa apenas filtrar os pacotes com o valor de PID utilizado e extrair de sua carga os dados transmitidos. Esse mecanismo é utilizado para o transporte de dados assíncronos, não utilizando nenhum mecanismo de temporização. Vale ressaltar que a organização e a formatação dos dados dependem completamente da aplicação.

3.1.2 Data Streaming

Este mecanismo oferece suporte para serviços que requerem transmissão de dados síncronos ou assíncronos [10] [14]. No *data streaming*, os dados são continuamente empacotados em *Packetized Elementary Streams* (PES) antes de serem difundidos. O suporte a apresentação de dados de maneira síncrona é oferecido com o uso de marcadores de tempo no cabeçalho do PES. Um pacote PES pode carregar, aproximadamente, 64 Kbytes de dados e marcadores de tempo para sincronização. Dois dados de sincronismo, inseridos no cabeçalho dos PES, são utilizados para auxiliar o receptor durante a apresentação do conteúdo transmitido: o *Presentation Time Stamp* (PTS), que indica o instante em que a apresentação deve ser realizada, e o *Decode Time Stamp* (DTS), que indica o instante de término da decodificação [12].

No *data streaming*, também é possível utilizar seções privadas para se realizar o encapsulamento dos dados. A sintaxe das seções privadas permite que estas sejam numeradas, permitindo uma contigüidade na inserção e filtragem dos dados. O mecanismo de *data streaming* também é uma forma de envio de dados com um baixo *overhead* (dados auxiliares e de cabeçalho). As seções permitem o uso de dois tipos de cabeçalhos: o cabeçalho simples e o estendido. Através da configuração de cabeçalho simples, são necessários apenas 3 bytes para cada seqüência de 4093 bytes de dados; utilizando-se o cabeçalho estendido, podem ser transmitidos até 4084 bytes de dados por seção.

Este mecanismo é indicado para o transporte de dados síncronos e assíncronos e o seu uso restringe-se apenas ao transporte de dados difundidos para todos os receptores (*broadcast*) de uma única vez. Para transportar dados para um grupo de usuários, é necessária a utilização um mecanismo com suporte a protocolos de rede.

Um exemplo de *data streaming* é a transmissão de *Closed-caption* em redes de TVD.

3.1.3 Multiprotocol Encapsulation (MPE)

O MPE oferece um mecanismo para transportar datagramas com o TS MPEG2. Ele pode ser utilizado para transportar qualquer tipo de protocolo de rede com o uso do encapsulamento LLC/SNAP⁴, cobrindo *Unicast* (datagramas enviados para um único receptor), *Multicast* (datagramas enviados para um grupo de receptores) e *Broadcast* (datagramas enviados para todos os receptores). Um endereço *Media Access Control* (MAC) de 48 bits é utilizado para endereçar receptores [9] [14].

Os sistemas de difusão apresentados até aqui satisfazem a maior parte das necessidades de transporte de dados para as soluções proprietárias. Porém, a tecnologia de TV Digital também faz uso de mecanismos mais complexos, como é o caso dos Carrosséis, que transportam dados de forma cíclica e estão associados ao uso do *middleware*.

3.1.4 Carrosséis

Carrosséis são protocolos de difusão de dados definidos pelo padrão DSM-CC (*Digital Storage Media, Command and Control*). O nome deriva do fato deste protocolo permitir uma repetição cíclica, de um determinado conjunto de dados, em um fluxo de transporte. Essa característica auxilia o receptor no acesso aos dados, visto que o mesmo, em busca de determinada informação, necessita apenas aguardar sua próxima repetição. Também é possível, a partir do receptor, adotar mecanismos de armazenamento prévio de conteúdo de um carrossel, de forma a agilizar a disponibilidade dos dados à alguma aplicação. Maiores detalhes sobre este mecanismo podem ser vistos em [9] [14].

3.1.4.1 Carrossel de dados

O carrossel de dados é o mecanismo mais simples definido pelo DSM-CC e proporciona uma organização lógica baseada em módulos, que são então associados em grupos. Como os dados são encapsulados nessas estruturas, esta forma de *datacasting* é fortemente recomendada para a difusão de dados divididos em unidades de tamanho determinado, porém, não há nenhuma descrição sobre qual o tipo do dado que é transportado por um módulo e o seu correto tratamento é de responsabilidade da aplicação. O carrossel de dados permite também que determinados módulos sejam difundidos em intervalos regulares de tempo menores que outros [9] [14].

3.1.4.2 Carrossel de objetos

Os carrosséis de objetos baseiam-se nas definições dos carrosséis de dados, porém, passam a tratar a informação na forma de objetos, que apresentam uma estrutura associada. Para a difusão de dados, dois objetos são de grande importância em um carrossel deste tipo: objetos do tipo *arquivo* e do tipo *diretório*. Com esses dois tipos, é possível formar um sistema de arquivos simples. Dessa forma, o terminal de acesso pode reter arquivos de um sistema de arquivos que está sendo difundido em um carrossel de

⁴ LLC/SNAP (*Logical Link Control / Subnetwork Access Protocol*)

objetos, como se os mesmos estivessem disponíveis localmente. Devido a essa característica, o carrossel de objetos pode ser definido como um sistema de arquivos de difusão [9] [14].

3.1.5 Avaliação dos mecanismos de Datacasting

Para a avaliação do método de *datacasting* mais adequado à transmissão do feixe de bits utilizado na reconfiguração do FPGA, foi levado em consideração o formato dos dados e os requisitos temporais (sincronização) associados ao dado transportado.

Com relação ao formato, os dados são classificados em: delimitados, não delimitados e datagramas. Os dados delimitados são aqueles que podem ser divididos em unidades de tamanho determinado, como em objetos, arquivos, etc. Por outro lado, os não delimitados são considerados como um fluxo contínuo de bits. Por fim, os datagramas correspondem à fragmentação dos dados em pacotes, seguindo-se algum protocolo de comunicação [13]. Quanto aos requisitos temporais, os dados são divididos em síncronos, sincronizados e assíncronos. Os dados síncronos possuem requisitos de sincronização com outros dados do mesmo fluxo, conhecido por sincronização intra-mídia. Dados sincronizados são aqueles que devem ser apresentados em instantes pré-determinados de tempo e em sincronismo com elementos de outras mídias como, por exemplo, fluxos de vídeo. Esse tipo de sincronismo é denominado inter-mídia. Os dados assíncronos, por sua vez, não possuem requisitos temporais de sincronização (Tabela 2).

Tabela 2: Características dos mecanismos de datacasting.

Requisitos	D a t a p i p i n g	Data streaming		C a r r o s s é i s	M P E
		Seções Privadas	PES		
Suporte a detecção de erros	-	x	x	-	x
Suporte a outros tipos de protocolos	-	-	-	-	x
Suporte a unicast	-	-	-	-	x
Suporte a multicast	-	-	-	-	x
Suporte a broadcast	x	x	x	x	x
Suporte a dados não delimitados	Síncronos	-	-	x	-
	Assíncrono	x	x	-	-
	Sincronizado	-	-	x	-
Suporte a dados delimitados	Síncronos	-	-	-	-
	Assíncrono	-	x	-	x
	Sincronizado	-	-	-	x
Suporte a datagramas	Síncronos	-	-	x	-
	Assíncrono	-	-	-	x
	Sincronizado	-	-	x	-

“x”, método mais adequado. “-”, não tem suporte ou menos adequado.

Para efetuar a reconfiguração do hardware (atualização do decodificador de H.264), que é o foco deste trabalho, os dados transportados são considerados delimitados. O feixe de bits (com o conteúdo de reconfiguração) poderá ser dividido em um número finito de fatias de tamanho determinado, antes de ser carregado em um mecanismo de transporte.

Do ponto de vista dos requisitos temporais, os dados são assíncronos, pois o feixe de bits não precisa de nenhuma sincronização com outros tipos de dados ou elementos de outras mídias. A preocupação reside em recuperar os dados de forma

segura e, se possível, com a utilização de algum mecanismo de detecção de erro.

Utilizando-se os carrosséis, há uma grande facilidade na recuperação da informação transportada, tendo-se em vista que esse mecanismo já tem suporte à repetição cíclica, facilitando a recuperação da informação pelo receptor. Porém, os carrosséis, por utilizarem estruturas complexas, também necessitam de mecanismos de encapsulamento mais elaborados durante o processo de multiplexação do TS. Na recepção do conteúdo, as estruturas computacionais necessárias para se extrair as informações das estruturas utilizadas pelo carrossel são complexas e conseqüentemente consomem mais recursos computacionais e espaço em memória.

O mecanismo de *data streaming* via seções privadas é mais simples e tem suporte à detecção de erros, garantindo a segurança do sistema, além de oferecer suporte a dados assíncronos e delimitados e utilizar estruturas menos complexas que os carrosséis. Com a utilização do mecanismo de *data streaming* via seções privadas, o feixe de bits (dado delimitado) pode ser quebrado em seções de até 4080 bytes mais os 4 bytes do CRC. As seções são numeradas de acordo com a ordem de inserção das fatias (cada 4080 bytes de dados) do feixe de bits. Esse mecanismo também permite configurar a taxa de repetição das seções. Desta forma, a aplicação residente no receptor consegue recuperar a informação enviada, além de manter a ordem original dada no processo de encapsulamento.

Dentre os dois mecanismos mais adequados ao transporte dos dados, o *data streaming* via seções privadas é o escolhido. Com relação aos critérios de escolha que foram levados em consideração, é possível destacar a quantidade de recursos computacionais necessários, a complexidade de implementação e a complexidade de empacotamento (multiplexação) do TS.

4. PLATAFORMA RECONFIGURÁVEL

4.1 Plataforma de STB

O STB (ou plataforma) é um dispositivo de recepção de sinais de televisão digital conectado a um televisor por meio de cabos ou qualquer outro tipo de conexão e que, para tanto, disponibiliza interfaces de saída de áudio e vídeo, sejam estas analógicas ou digitais [2]. A configuração básica do receptor está ilustrada na Figura 3 e é composta pelas seguintes unidades: a) antena de recepção terrestre, b) *Integrated Receiver Decoder* (IRD) e c) saídas de áudio e vídeo.

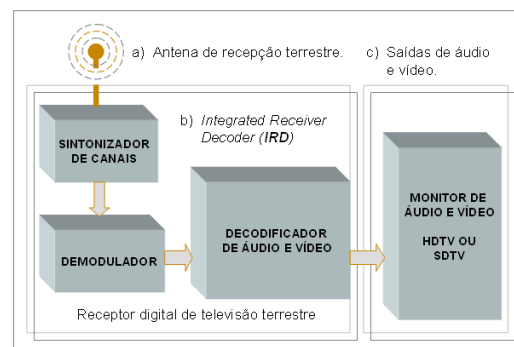


Figura 3: Configuração básica do receptor, Fonte: [2].

Grande parte dos STBs existentes no mercado contém arquiteturas de hardware similares. Essas plataformas são geralmente compostas por um processador de propósito específico, que gerencia as interações entre os diversos módulos de hardware da

arquitetura do STB, uma memória de armazenamento persistente (e.g. FLASH) para o sistema de arquivos, as configurações de sistema e programas residentes e uma memória volátil (e.g. SRAM) para a carga e execução de software. Além disso, há outros componentes como demoduladores, demultiplexadores, front-end, controladores de vídeo, áudio, etc.

No processamento realizado pelos diversos módulos em uma arquitetura de STB, encontram-se algumas partes implementadas em software e outras em hardware. Na plataforma de referência tomada por base para este trabalho, o decodificador de vídeo H.264 está implementado em um núcleo (DSP – *Digital Signal Processor*) integrado ao circuito do processador principal (Figura 4). O DSP é responsável pela decodificação de vídeo, cuja saída é então enviada ao codificador de saída de vídeo (*Digital Encoder* – DENC, que disponibiliza sinais no padrão do país), até chegar à conversão digital-analógica, e às saídas digitais (e.g. HDMI). O *front-end*, representado pelo Tuner (sintonizador, composto por PLL e demodulador) e dispositivos periféricos, efetua a recepção do sinal transmitido (interface aérea) e gera como saída um TS.

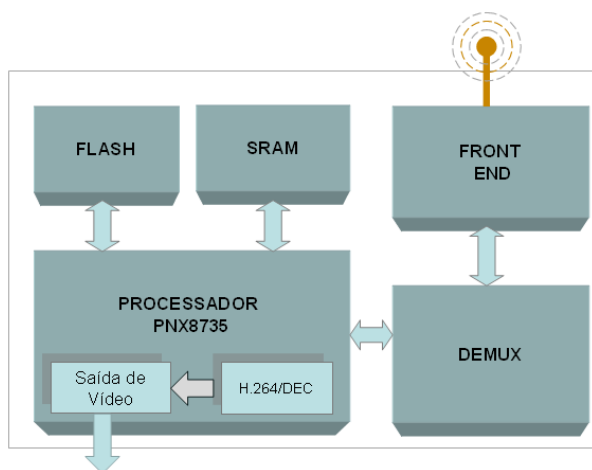


Figura 4: Arquitetura base do STB.

Após a saída do *Front-end*, o TS é demultiplexado, separando-se áudio, vídeo e dados. O DEMUX filtra o TS, através de filtros baseados em PID, enviando os pacotes obtidos aos seus respectivos módulos de processamento.

4.2 A plataforma STB reconfigurável

Para a demonstração do paradigma de uma plataforma de STB reconfigurável através do sinal de TV Digital, foi considerada a atualização do decodificador de vídeo H.264. Para a realização desse processo, é necessária a integração de um circuito de FPGA e uma memória FLASH à arquitetura de STB mencionada. Esta nova arquitetura está ilustrada na Figura 5. Na nova abordagem, a saída do filtro de PID de vídeo é encaminhada ao circuito de FPGA através de uma interface padrão disponibilizada pelo processador principal (e.g. DVB-CI – *Common Interface* ou USB 2.0). Após a decodificação, o conteúdo de vídeo recuperado é retornado ao processador e novamente inserido na cadeia de processamento de vídeo, para ser reescalonado, combinado e disponibilizado através das interfaces de saída.

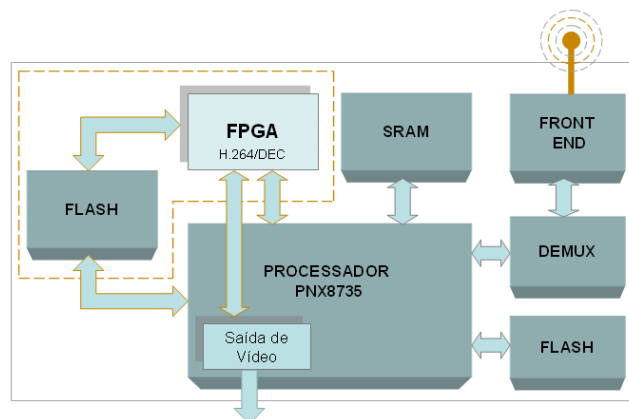


Figura 5: Arquitetura com o FPGA integrado.

Nesta arquitetura, o FPGA tem comunicação direta com o processador, a cadeia de processamento de vídeo (interna ao processador) e a memória FLASH. O processador e o software de controle implementam a filtragem de seções/tabelas em busca do conteúdo de atualização. Para se realizar este trabalho de forma sistemática, cria-se uma nova tabela chamada de Tabela de informação de atualização (UIT – *Update Information Table*), cujo objetivo é informar uma atualização de software relativa ao decodificador de vídeo e o PID dos pacotes que carregam o fluxo de bits de reconfiguração. Após a obtenção desta tabela, o PID do feixe de reconfiguração é informado, o que então permite o download completo do fluxo de bits pelo receptor, através de um filtro de PID localizado no DEMUX. O download é feito diretamente na memória FLASH (FPGA), seguido de uma remontagem dos dados para uma organização correta do arquivo de reconfiguração. Sendo assim, com o binário gravado corretamente em memória, o processador reinicia o sistema baseado em FPGA, o que permite o funcionamento do novo decodificador.

A atualização do decodificador de H.264 é apenas um exemplo prático de aplicação da reconfiguração de hardware utilizando STBs e o sinal de TV Digital. No entanto, esta mesma metodologia pode ser utilizada para a atualização de outros componentes e também no desenvolvimento de novas aplicações.

5. CONSIDERAÇÕES FINAIS

Os resultados esperados neste projeto estão diretamente ligados ao contexto de sua aplicação. Em um sistema de TVD, este trabalho pode contribuir através da redução do legado existente na implantação de um novo padrão de TVD ou na melhoria de um existente. Com o avanço da tecnologia, os receptores poderiam ser simplesmente atualizados com diferentes tipos de decodificadores, demoduladores, etc. Por exemplo, um decodificador H.264 poderia ser programado com uma versão mais recente sugerida numa revisão da norma de TVD, o que possibilitaria à emissora explorar todas as facilidades resultantes do novo algoritmo muito mais rapidamente, sem a necessidade de esperar que o usuário adquira um novo aparelho. Com isso, os custos globais envolvidos no processo também seriam consideravelmente reduzidos. Entretanto, tal prática ainda dependeria de um contato próximo entre emissoras e fabricantes de receptores, de modo a se atualizar todos os dispositivos existentes na rede de TVD.

6. AGRADECIMENTOS

Agradeço ao CT-PIM (Centro de Ciência, Tecnologia e Inovação do Pólo Industrial de Manaus) pelo apoio prestado para a realização deste trabalho.

7. REFERÊNCIAS

- [1] ABNT NBR 15602-3, Norma Brasileira. Televisão digital terrestre - Codificação de vídeo, áudio e multiplexação, Parte 3: Sistemas de multiplexação de sinais. 2007.
- [2] ABNT NBR 15604, Norma Brasileira. Televisão digital terrestre - Receptores, 2007.
- [3] Actel Corporation, "Flash FPGAs in the value-based market white paper," Tech. Rep. 55900021-0, Actel, Mountain View, Calif, USA, 2005. <http://www.actel.com>.
- [4] Agostini, L. V., Porto, M., Güntzel, J. L., Porto, R. C., Bampi, S. High Throughput FPGA Based Architecture for H.264/AVC Inverse Transforms and Quantization. In: MWSCAS 2006 - 49th IEEE International Midwest Symposium on Circuits and Systems, San Jose, 2006.
- [5] Agostini, L. V., Azevedo, A., Staehler, W., Rosa, V., Zatt, B., Pinto, A. C., Porto, R. C., Bampi, S., Suzin, A., Design and FPGA Prototyping of a H.264/AVC Main Profile Decoder for HDTV. Journal of the Brazilian Computer Society, v. 13, p. 25-36, 2007.
- [6] Altera, "Standard Cell ASIC to FPGA Design Methodology and Guidelines," Abril de 2009. <http://www.altera.com>.
- [7] Altera, "Supporting Digital Television Trends with Next-Generation FPGAs," Abril de 2010. <http://www.altera.com>.
- [8] ETSI, "Digital Video Broadcasting (DVB): Framing structure, channel coding and modulation for digital terrestrial television - V1.5.1," ETSI standard, EN 300 744, 2004.
- [9] ETSI, Digital Video Broadcasting (DVB); Implementation guidelines for Data Broadcasting, ETSI standard, TR 101 202, V1.2.1, 2003.
- [10] ETSI, "Digital Video Broadcasting (DVB): Specification for Data Broadcasting," ETSI standard, EN 301 192, V1.5.1, 2009.
- [11] Garcia, P., Compton, K., Schulte, M., Blem, E., Fu, W., An Overview of reconfigurable hardware in embedded systems, EURASIP Journal on Embedded Systems, v.2006 n.1, p.13-13, 2006.
- [12] ISO/IEC 13818-1, "Information technology - Generic coding of moving pictures and associated audio information-Systems", 2007.
- [13] Piccioni, C. A., Modelo e Implementação de um Serviço de Datacasting para Televisão Digital. Dissertação de Mestrado, Universidade Federal de Santa Catarina, 2005.
- [14] Reimers U., DVB - The Family of International Standards for Digital Video Broadcasting. New York: Springer-Verlag, 2004.
- [15] Xilinx, "Comparing and Contrasting FPGA and Microprocessor System Design and Development," - 2004. <http://www.xilinx.com>.