Hardware Decoding Architecture for H.264/AVC Digital Video Standard

Alexandro C. Bonatto, Henrique A. Klein, Marcelo Negreiros, André B. Soares, Letícia V. Guimarães and Altamiro A. Susin

Department of Electrical Engineering
Federal University of Rio Grande do Sul, Brazil
1 Introduction

Nowadays, embedded computing is extensively used in electronic devices for to capture process and display digital videos. There was a fantastic evolution from the first television transmission systems till the multi format, multi view and high definition television available today. This chapter will explore several techniques underlying the Digital Television (DT) that made it possible to capture, process, store and broadcast digitized moving pictures. The focus of this chapter will be the Image Processing to compress and decompress video sequences, since this is a critical part of a digital television system. A digitized movie contains also sound and data information which are coded aggregated to the video to form what is called the video stream.

The evolution of digital television happened at the same pace of the technological progress, particularly in the electronics domain. In fact, only the huge amount of processing power available in VLSI (Very Large Scale Integrated) circuits enables the implementation of sophisticated algorithms that run in digital television apparatuses. The amount of raw video information to display on a high definition television screen is too big to be transmitted or stored at a reasonable cost. Coding the raw video information in a digital domain allows the video compression, maintaining image quality and reducing the amount of data. With sufficient processing power it is possible to explore the spatial and temporal redundancy that exists in a moving image and reduce more than two orders of magnitude the amount of data to be transmitted. In order to reconstruct the movie a powerful digital system is employed at the receiver side. Nowadays we can afford such system at a cost that made them accessible to everyone. Like that, analog television systems are being replaced by its digital counterpart all over the world.

The techniques to compress video are normalized by standards. This chapter will consider the newest standard, the H.264/AVC, also known as MPEG4 part 10 Advanced Video Coding (AVC). The algorithms defined in this standard are discussed in this chapter and the architecture of the developed digital system that implements these algorithms is detailed. The H.264/AVC standard doubles the video compression if compared to its precursor (MPEG2) on the same video quality. Therefore it occupies a smaller bandwidth for transmission, reducing the required storage space. Similarly this advance in compression allows increasing the video picture size while maintaining image quality. The bandwidth reduction makes the H.264/AVC an excellent choice for digital television broadcasters who want to distribute content on High Definition Television (HDTV) or to reduce the cost of carrying conventional Standard Definition (SD) channels. Terrestrial television broadcasting is one of the most popular information spreading mechanisms and has reached several social classes for the last eighty years. It has received several enhancements since its first prototype, and the most recent one was the transition to digital terrestrial television, which is being still implanted in several countries. DT represents a new way of accessing information, enabling the transmission of different types of programs and allowing the use of interactivity between the television station and the viewer. Nowadays, in Brazil the terrestrial television broadcasting system is under gradual transition from the analog PAL-M standard to the ISDB-T (Integrated Services Digital Broadcasting - Terrestrial), also called SBTVD (Brazilian Digital Television System). ISDB-T Standard was adopted by Japan in 2003 and in Brazil in 2007, after receives important technological improvements. Besides promoting the study and development of state-of-the-art technologies, the ISDB-T has adopted H.264/AVC as the standard for broadcasting digital video (ABNT, 2007). This video coding standard is also being included in other terrestrial DT standards like DVB-T and ATSC, as presents Table 1.
Broadcasting is the distribution of audio and video content using radio waves. The Set-Top Box (STB) receives the analog signal carried by the electromagnetic waves in the antenna and extracts digital video, audio and data information. The system’s core is the media processor which can handle complex processing functions such as decoding, resizing and transrating (digital-to-digital data conversion to a lower bit rate without changing the coding standard) media streams. The implementation of a digital television STB requires high performance hardware for video and audio decoding and demodulation. Moreover, the DT receiver must have a low cost to be widespread, mainly integrating its main modules into a single SoC (System-on-Chip), thus eliminating the use of several discrete chips. This enhances the electronic system's reliability reducing the receiver dimensions, the input and output pin count and therefore its project complexity.

Motivated by the adoption of the digital TV standard in Brazil, an effort has been devoted to develop dedicated hardware architectures for a H.264/AVC Standard video decoder. The H.264/AVC is nowadays the most advanced video coding Standard of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). This state-of-the-art standard outperforms previous standards by employing bi-predictive motion estimation, spatial prediction and adaptive entropy coding techniques. However, this compression improvement has as penalty an increase in computational demand and in memory traffic when compared to previous standards. The design and validation of the set-top box require a big effort from the design team due to its processing complexity. Dedicated modules must supply the high performance demand of the decoding processing. Memory access optimizations are also necessary to handle data in the reference frames while video is decoded and exhibited in real time. Besides the multimedia decoding, this system requires to manage the displayed video and compose it with text and graphics data, referred as the On-Screen Display (OSD) information. Furthermore, the format of the video output must be scaled to fit the video display format. Video post processing operations are done in the video using digital signal processing techniques.

<table>
<thead>
<tr>
<th>Standard</th>
<th>Location</th>
<th>Modulation</th>
<th>Video Standard</th>
<th>Audio Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVB-T</td>
<td>Europe, Asia, Africa</td>
<td>OFDM based</td>
<td>MPEG2/H.264</td>
<td>AC3/AAC/other</td>
</tr>
<tr>
<td>ATSC</td>
<td>North America</td>
<td>8VSB</td>
<td>MPEG2/H.264</td>
<td>AC3</td>
</tr>
<tr>
<td>ISDB-T</td>
<td>South America, Japan</td>
<td>OFDM based</td>
<td>H.264/MPEG2</td>
<td>AAC</td>
</tr>
<tr>
<td>DTMB</td>
<td>China</td>
<td>OFDM based</td>
<td>AVS (Audio Video Standard)</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Terrestrial Digital Television Standards.

2 Digital Video Processing Background

This section presents the sufficient background information underlying the context and significance of digital video processing algorithms. A video sequence looks like a continuous motion, but it is formed by a series of still images that change fast enough to deceive the human eye. A picture element (pixel) is the smallest part of an image and a rectangular array of pixels forms an entire image. A higher visual quality is achieved with more pixels on the image. Also, increasing capture and exhibition rates of video images (frames per second) leads a sensation of real motion.
Digital videos can be represented using different color formats. When a sequence of pictures is captured by a camera, all information in the physical world is translated to the digital domain using sampling and quantization. Modern video cameras contain image sensors with millions of pixels. In grayscale systems, each pixel represents single information but in color systems, a pixel represents a quantity of luminosity and a quantity of color that can be represented in different color spaces. Pixel information can be coded using the fundamental colors in RGB (red, green and blue) color space or using luminance and chrominance in the YCbCr color space: luminance (Y), chrominance blue (Cb) and chrominance red (Cr). The human eye is less sensitive to color information than luminous information, thus video systems usually represent the pixel information using YCbCr color space because it facilitates the subsampling of color information when compared to RGB. Pixels represented in the YCbCr color space can be subsampled in order to reduce the information needed to store an image by up to 50%. High quality pictures are represented in 4:4:4 format, while 4:2:0 is used in video systems to compress images by a factor of 2 with a 4:1 subsampling on each chrominance component. Also, in digital television high definition video adopts YCbCr ITU Recommendation 709 instead of standard definition videos which use YCbCr ITU Recommendation 601.

Video sizes are increasing with the evolution of the image sensors and capture devices for video systems, consequently the system bandwidth increases as well. Pixel bandwidth requirements between the image sensor and the processing system increase with the picture size, frame rate and pixel bit depths. A video sequence is formed by still images captured in different time instants. Picture size is defined by the number of vertical pixels times the number of horizontal pixels. Frame rate refers to the number of pictures per second that are taken to make a video sequence. The standard in television systems is 30 frames per second, using interlacing technique to improve the appearance of motion. In most consumer video systems the images changes occurs 50 or 60 times per second. Table 2 shows digital video rates for different resolutions used in DT for RGB and YCbCr color spaces, considering 8 bits per pixel color component.

<table>
<thead>
<tr>
<th>Video Resolution (width x height)</th>
<th>Video Format</th>
<th>Pixel Rates (Mb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RGB</td>
</tr>
<tr>
<td>320x240</td>
<td>QVGA</td>
<td>55,3</td>
</tr>
<tr>
<td>720x480</td>
<td>SD</td>
<td>248,8</td>
</tr>
<tr>
<td>1280x720</td>
<td>HD</td>
<td>663,6</td>
</tr>
<tr>
<td>1920x1080</td>
<td>Full-HD</td>
<td>1493,0</td>
</tr>
</tbody>
</table>

Table 2: Digital video pixel rates for display 30 frames per second.

### 2.2 Digital Image Block Partitioning and Processing

The processing of digital images is done in small rectangular blocks of pixels, grouped by the pixel neighborhood, in the form of regions of different sizes. Most common regions of pixels used in image processing are: 4x4, 8x8, 8x16, 16x8 and 16x16 pixels. The last is called a macroblock of pixel samples, or simply macroblock. A group of macroblocks is called a slice. Processing video is done by
manipulating the pixels in coarse-grain tasks, as quantizing, filtering, transforming, that are mapped onto separated processing units. Also, a common definition used in video processing is the Line-of-Pixels (LoP) that represents a group of 1×4 pixels. The order as the pixels are processed in the macroblock is maintained in every coding or decoding process. The raster scanning of an image is convenient for exhibition, as used by monitors or in television. In the macroblock processing the H.264/AVC standard adopts the double-Z order, as shown in the Figure 1. Also, each macroblock pixel sample must be represented by its luminance (luma) and blue/red chrominance (chroma) components. Thus, different pixel processing sequences can be adopted in an architectural design of video coding systems, from separated hardware unit for processing luma and chroma components to a single unit multiplexing in time the element processed. A macroblock represented in YCbCr 4:2:0 consists of a sequence of 256 luminance pixels followed by two 64 chroma pixels sets.

![Figure 1: Pixel array organization in a macroblock of 16×16 pixels.](image)

2.3 H.264/AVC Video Image: Frame and Field Pictures

A video is formed by a sequence of still images displayed in a sufficient high rate to create the illusion of continuous motion. The video display emits light just in a fraction of the image exhibition time and refreshes to light again the image. Human vision cannot perceive the display flicker effect if the refresh rate is higher than 48Hz (Poynton, 2007), like in cinema. In television systems the typical refresh rate is higher than 60Hz. The video picture rate is independent of the display refresh rate and can be slower than it. Typically in television systems the picture (or frame) rate can be: 60Hz, 50Hz, 30Hz, 25Hz and 24Hz. Video scanning is done capturing pixels in a sequential order and determines a fixed relationship between the pixel position in the image and the time instant. The video sequence can be captured and encoded as frame or field pictures. When the image is captured as a frame picture it is sourced in a progressive way. The process of scan an image called raster scan is done in a uniform rate, from the left to right pixels in the image line and from the top to the bottom image lines. A frame contains an array of luma samples and two corresponding arrays of chroma samples. When the image is captured as a field picture it is sourced in an interlaced way. The information gathered in each time interval is cut by half by acquiring
alternately even and odd lines of pixels of the image, as shown in Figure 2, alongside the comparison between a capture in frame picture mode.

Considering two consecutive captures in field picture mode, the first capture consists of the even lines of the scanned image and is designated as the top field while the second capture is called bottom field and contains the odd lines of the scanned image (Figure 3). To encode a video sequence it is possible to choose between either progressive or interlaced modes. Choosing to encode in an interlaced way, these fields can be encoded in one of two ways: frame mode of field mode. The frame mode interlace both fields to form an interlaced image, while the field mode separates them placing the top field on the upper half of the image and the bottom field on its lower half, both modes are depicted in Figure 3.

**Figure 2:** Image capture modes in different time intervals $t_1$ and $t_2$ for field mode (interlaced) and frame mode (progressive).

**Figure 3:** Types of encoding of a field picture capture.
An interlaced video sequence can be encoded in either frame or field mode. The frame mode combines two fields into a frame by using even lines from one field and odd lines from another field (see Figure 3b). The field mode combines two fields into a frame by copying all lines from the first frame and then all the lines from the second frame (see Figure 3a).

The decision of which mode the field picture will be encoded is called Picture Adaptive Frame Field (PAFF), as the decision method is based on each image. However the process of choosing the encoding mode can occur within the picture, in the macroblock layer, therefore called Macroblock Adaptive Frame Field (MBAFF). In this case, the notion of grain is a region of 16×32 pixels, i.e. two macroblocks of 16×16 pixels vertically connected, where each 16 lines are from two consecutive fields, as shown in Figure 4. The MBAFF coding enables each pair of macroblocks to be encoded either using frame or field mode, resulting in numerous possible configurations in a single picture. An example of MBAFF coding is shown on Figure 5.

2.4 Digital Television Transport and Program Streams

The digital multimedia information that is broadcasted by terrestrial television stations and received in the set-top boxes is carried in data packets according to ISO/IEC 13818-1 and ITU-T H.222.0 Standards.
The Elementary Stream (ES) is the raw output of an encoder, containing only compressed audio or video data. The ES can be broken into smaller size data blocks, called Packetized Elementary Stream (PES). The PES divides the ES into packets, with the addition of a header containing information regarding the content of the ES and also time stamps that can be used for synchronization.

The combination of PES with the addition of a synchronization method and a program clock reference, in order to allow the correct use of the PES time stamps, form the Transport Stream (TS). The transport stream can carry a single program (SPTS) or multiple programs (MPTS). Additional control information is also contained in the TS, in order identify how many programs there are in the TS, how to link each of the programs to their corresponding elementary streams, and other information like Electronic Program Guide. TS consists of 188 bytes fixed-size data packets.

The next Section presents the H.264/AVC Standard in a comparison with the ISDB-T Standard. Also it is shown a hardware architectural design and implementation of the video decoder to be prototyped using programmable logic devices or to be implemented in silicon.

3 The H.264/AVC Standard and Hardware Architectural Implementation

3.1 Advanced Video Coding Standard

The H.264 Advanced Video Coding Standard defines a set of levels and profiles, to adapt the coding process in different video resolutions, reducing pixel rates and visual characteristics of a picture sequence. Data processing capacity and output bandwidth are defined by the level, while profiles refer to different set of coding functions. There are several different coding profiles defined by the Standard as: low power and low cost applications (baseline profile), scalable, multiview and high quality picture coding (high profile). The main profile includes television broadcasting and video storage, supporting interlaced videos, inter-coding with bi-predictive slices, inter-coding with weighted prediction and entropy coding using Context-Adaptive Arithmetic Coding (CABAC). The ISDB-T International set-top box specification and characteristics are standardized in Brazil by the ABNT (2007) Standard. The same occurs for the transmission, multiplexing, and interactive channels among others. The ABNT Standard for video coding and decoding is based on the ITU-T H.264/AVC standard but with restrictions to some H.264/AVC features. Because the ISDB-T Standard was firstly adopted in Japan and the referred standard was adopted and modified in Brazil, reference documents are available by DiBEG (Digital Broadcasting Experts Group) for characterizing specifications between the Brazilian and Japanese Terrestrial DT Standards (DiBEG, 2009).

The decoding process in the H.264/AVC digital video coding standard (ITU-T Recommendation) is described in six sub-sections: (1) NAL unit decoding; (2) Slice decoding process; (3) Intra prediction process; (4) Inter prediction process; (5) Transform coefficient decoding process and picture construction process prior to deblocking filter process and; (6) Deblocking filter process. Figure 5 shows the video reconstruction path in an H.264/AVC decoder, from the compressed video bitstream to the current frame in exhibition. The decoder is formed by three main parts: residual generation, prediction generation and the final image generation. Predicted macroblocks reconstructed from reference frames of from previously decoded macroblocks (Intra prediction). Residuals are combined to the prediction blocks and filtered before the exhibition and storage in the decoded picture buffer.
Figure 6: The H.264/AVC decoding process represented by its main algorithmic functions connected through arrows, which represent the flow of data between processing blocks.

The compressed video bitstream is received in the video decoder within the Network Abstraction Layer (NAL) unit. The entropy decoder contains a Parser that receives a compressed video bitstream in the input from the NAL and decodes the quantized coefficients to generate the residual data. Also, it extracts the syntax elements to inter-frame and intra-frame prediction processes. The residual data is decoded using fixed or variable length binary codes in one of the entropy decoders: Exp-Golomb, CAVLC or CABAD decoder. Also, the residual is processed in the inverse transform and inverse quantization (IT and IQ) steps. Using information decoded from the bitstream, the decoder creates a prediction block. The H.264 Standard adopts two modes of block prediction: intra and inter prediction. Inter prediction refers to the reuse of information previously decoded in past or future pictures, stored in the decoded picture buffer. Intra-prediction reconstructs each image block from its neighborhood. Finally, the residual data is added to the predicted blocks of pixels, generating the pixel output that is filtered before exhibition.

3.2 The Network Abstraction Layer Unit

The Network Abstraction Layer Unit (NALU or NAL unit) is an element of the Elementary Stream (ES) that contains the compressed video data. The video elementary stream is part of the transport stream TS described in Section 2.4. There are 12 types of NAL units and each of them carries specific information. A NAL unit is a data packet whose limits are identified by a start_code_prefix code, as shown in Figure 7. The start_code_prefix is a sequence of 3 consecutive bytes. When the decoder receives the start_code_prefix in the input ES, it detects that the current NALU has ended and that a new NALU has started.

Figure 7: Start_code_prefix and NALUs location.
Each NALU contains two parts: header and data. The NALU header presents the NALU identification as is shown in Figure 8. There are three classes of NALU: (1) Video Coding Layer (VCL), (2) Parameters and, (3) Control.

![Figure 8: Structure of a NALU.](image)

Because the video processing is done in macroblocks of pixels, the video decoder must be configured for process each macroblock in a specific way. This configuration is carried in the NALUs, together with the coded residual and prediction information. Important configuration parameters are stored in NALU type 5 (IDR, Instantaneous Decoding Refresh) with the parameters contained in NALU type 7 (SPS, sequence parameters) and NALU 8 (PPS, picture parameters). However, if the slice is a typical P or B type (inter frame prediction modes), stored information is needed for decoding them and the decoder uses a configuration parameters obtained from previous decoded pictures (for P and B type) and forward decoded picture (for B type).

### 3.3 H.264/AVC Video Decoder Hardware Design and Implementation

To implement a digital system that executes the tasks defined by an algorithmic description is not an obvious work. There are many parallel processes and many ways to implement them and this raises the system complexity. The H.264/AVC Standard defines the video coding behavior by functions and tasks, explaining what the system must do when decoding video signals. The standard does not define the system implementation and how the process is done. The hardware architectural implementation of the digital circuit begins with the detailed analysis of the system behavior and algorithm description, usually making a model in a programming language such as C/C++. The algorithmic description of the H.264/AVC video decoder was partitioned into separate processing units, according to every function performed in the decoding process. These processing units would be mapped to hardware blocks. Also, the architectural definition is based on information provided by the H.264/AVC Standard and also the H.264/AVC JM Reference Software from the Joint Video Team (JVT 2009).

Traditionally, there are two hardware design approaches that are used in the development of complex systems: the top-down and the bottom-up. In the top-down approach, the design process begins specifying the global system behavior, in a centralized approach, where the resources are accessible to every module in the system. The system specification is defined in terms of system state, in which a module should be able to estimate or retrieve resources that are local in other modules, within a known...
time delay. After the entire system specification and knowledge, the design effort is concerned to the implementation of the individual modules and intercommunication structure. However, in the bottom-up approach, the system is initially designed in terms of individual modules, designed and validated independently. Interaction between modules is modeled by the input and output ports behavior and the entire system state is unknown. In this project, the design team chooses the top-down approach for modeling the entire system using a C/C++ as mentioned above. The model is called the PRH264 (H.264 Reference Program) and defines only the functionality of the modules. The PRH264 does not define timing behaviors or communications structures between modules. The hardware was implemented in bottom-up design approach. Due to the high level of complexity of the whole system, each processing unit specified in the reference model was developed apart and integrated in an incremental way.

Based on the software model, the hardware modules were designed to implement different functions in the video decoding algorithm, and the video decoder design was segmented into Intellectual Property (IP) blocks. The IPs were described using VHDL language (VHSIC Hardware Description Language) to be implemented in programmable logic devices like FPGAs (Field Programmable Logic Arrays). The architecture of each module was defined targeting the video processing in 1920x1088 image sizes at a rate of 30 frames per second, considering different degrees of parallelism. Below is presented an overview of the architecture of each block. The hardware modules were independently developed by small sub-teams: Parser and Entropy Decoding; Intraframe Prediction; Interframe Prediction; Inverse Quantization and Transform; Filter and; DPB. After the IPs design phase, the team starts to integrate them to a single system, in different stages of integration. The H.264 video decoder architecture is shown on Figure 9.

Parser and entropy decoding is the first processing step and it is a highly sequential process. The second processing step (i.e. prediction and residual processing) enables a higher parallelism exploration between Intraframe and Interframe prediction and residual processing. Finally, prediction and residual are combined and filtered. Sample data (residual, prediction and image samples) flows in the system following the double-Z pattern, in LoPs.

**Figure 9:** The H.264/AVC decoding hardware architecture represented by its main processing units (PU). This digital system is composed by several PU developed separately and integrated to a single system.
The incoming coded video bitstream is composed by video and audio elementary streams, which are formed by packets called *Network Abstraction Layer Units* (NALU) as presented in Section 3.2. NALU processor is the hardware block in the parsers that identifies the NAL delimiters and feeds the Coded Video Buffer module, which in turn feeds the Parser module that interprets and decodes the NAL packets producing prediction and residual data to two FIFOs (First In First Out). The prediction data from one FIFO feeds the INTRA module. The coded residual data passes through CAVLD and together with the quantization step (QP) and the prediction block size data from another FIFO is sent to the ITIQ module. The INTRA module generates prediction information, while the ITIQ module generates the decoded residuals. Residual and predicted data are added to produce the reconstructed video, which is fed-back to the INTRA module to be used as a reference.

### 3.3.1 Parser and Entropy Decoding

The parser is the block that handles the compressed video bitstream within the video decoder as illustrated before in Figure 9. The parser is required to process the input bitstream, identify syntactic elements and route the associated data to the appropriate decoder module, like the inter-frame or the intra-frame prediction blocks. Syntactic element data may be encoded in different ways and an appropriate decoder module is needed. In a H.264/AVC video decoder the required decoders are called entropy decoders. The entropy decoders are Exp-Golomb decoder, CAVLD (*Context Adaptive Variable Length Decoder*) and CABAD (*Context Adaptive Binary Arithmetic Decoder*). The implemented hardware architecture for the parser supports the Baseline, Main and High profiles, and its block diagram is presented in Figure 10.

**Figure 10:** Parser architecture showing the syntactic element decoders and the required entropy decoders.

The Parser architecture is described in details in (Schmidt et al., 2011). The H.264 defines parameter sets containing information about the coded pictures. A Sequence Parameter Set (SPS) contains parameters that are applied to a complete video sequence as: the picture order count, decoded picture width and height and the choice of progressive or interlaced (frame or frame/field) coding. A
Picture Parameter Set (PPS) contains parameters that are applied to the current decoded picture as: an picture identifier, a flag to select VLC (Variable Length Coding) or CABAC entropy coding; the number the number of reference pictures in list 0 and list 1 that may be used for prediction, initial quantized parameters among others (Richardson, 2003). The Parser in the video decoder identifies the SPS and PPS in the received NALs to configure the decoder hardware units prior to decode Slice Headers and Slice Data.

The four corresponding decoding modules are enabled by the control module in a sequence determined by the data present on the bitstream:

- Prediction information is dispatched for the Intraframe Prediction Module: Macroblock type, Intraframe prediction modes and associated information;
- Interframe Prediction Module: Macroblock type, partition subtypes, Motion vector differences and reference image indexes;
- Residual information is dispatched for the Inverse Quantization and Inverse Transform Module: quantized transform coefficients;
- Image resolution is used in the DPB for image buffer sizing and in the video output for decoded picture exhibition.

### 3.3.2 Intra Frame Prediction Hardware Architecture Implementation

Intra prediction is a visual image compression process derived from the decoded samples of the same decoded slice. When coding a video picture, the macroblocks similarities in the same slice can be used to reduce information. The intra predictor is the coding process that represents a block of pixels (16×16 macroblock or 4×4 block) using previously coded blocks in the neighborhood of the current one. The video decoder generates blocks of pixels in intra mode (I frames) using previously decoded pixels of the current slice. The Intra architecture is shown on Figure 11a. It consists of four main parts: a context decoder, a neighbor fetching module, a prediction generator and a sample storage module.

The context decoder generates the prediction modes present on a macroblock, by using neighbors blocks modes and other information provided by the parser as the: decoding information regarding positioning of the macroblock and type of video (interlaced or progressive). In case of MBAFF coding, the encoding mode (field or frame) of each pair of macroblock is also decoded from the context.

The neighbor fetching module is provided with the decoded context and fetches the neighbors according to the type of video, positioning and mode of macroblock pair encoding. The prediction generator (shown with detail on Figure 11b) is provided with the neighbors and the prediction modes, which consist of 13 possible modes (ITU-T, 2005) used to calculate a weighted copy of neighbor image samples. The output of the prediction generator is composed by four predicted samples in parallel which are added to the residual samples outside the Intra-frame prediction module and fed back in to be stored by the sample storage module in dedicated local memories. The predicted samples stored internally will be used as reference for further samples, as well previous prediction modes, also stored in internal memories. Due to MBAFF coding, the memory dedicated to neighbor storage holds up to two lines of boarder pixels (as shown on Figure 12). The notion of macroblock pair encoding allows the neighbor of one macroblock to be within either a top field macroblock or a bottom field macroblock, therefore needing not only both neighbors samples but also a control to select which will be used.
Figure 11: Block diagram for the hardware architecture of the intraframe predictor (a) and a detail of the luma prediction module (b).
3.3.3 Inter Frame Prediction Hardware Architecture Implementation

In the video encoder, the inter prediction process creates a prediction model to the current encoded block from one or more previously encoded blocks, in block-based motion compensation search. The Motion Compensation (MC) process is used in the video decoder to find the current block from previously decoded blocks, stored in the decoded picture buffer. This work uses the MC hardware architecture implemented in (Zatt et al. 2007), composed by the main parts shown in Figure 13a: one motion vector prediction unit, one cache memory controller unit with external memory interface and two image samples processing unit. This MC hardware architecture supports both YCbCr 4:2:0 and YCbCr 4:2:2 color space formats (Zatt et al. 2008). The advantage of this architecture is the better use of memory locality, explored by a local cache memory of 32 sets of 40x16 pixels size. This cache memory is used to temporally store previously decoded macroblocks of pixels for reuse in the current macroblock generation. The advantage of using cache memories is extended to the video processing system by the improvement in global memory hierarchy performance, because of the frequency reduction of DPB accesses.

Motion Vectors (MV) generation is done in the motion vector prediction unit. They are used as an index to find the stored block in the DPB. After motion vectors generation and storage of the necessary blocks in the cache, the final process in the motion compensation module is to process the decoded block by image samples processing units. Separated luminance and chroma processing units are used in the separation of the Region of Interest (ROI) from the blocks stored in cache (based on the MV values) and for the interpolation of the ROI up to quarter pixel over the image samples. The separation of the ROI starts with the fetching of the necessary lines of pixel (each one is 40 pixels wide) from the cache for the interpolation process. After, a multiplexer separates only the necessary columns of pixels in the separated lines, producing the correct samples even when part of the ROI is outside of the image. The interpolation part of luminance processing unit is shown on Figure 13b. It is composed by two sets of FIR (Finite
Impulse Response) filters to generate half pixel samples, and bilinear interpolation units to produce quarter pixel samples. The first set of filters processes luma samples in the horizontal direction in parallel. The second set processes samples in the vertical direction in pipeline. The predictor is capable to generate four prediction image samples in parallel.

![Diagram](image)

**Figure 13:** (a) Motion compensation architecture; (b) detail of the architecture of the luminance component samples processing unit.

### 3.3.4 Deblocking Filter

The deblocking filter is the H.264/AVC video decoder output module used to remove block distortion that appears in the decoded pictures. This effect is generated because each video picture is segmented and processed in small macroblocks. After decoding, each macroblock is filtered by the deblocking filter, smoothing block edges and improving the appearance of displayed images. The Deblocking Filter architecture used in this work, developed by Rosa *et al.* (2007), is based on a 16 stage pipelined edge filter and it is shown on Figure 14a. It stores up to 4 macroblocks on an input buffer. On a first pass over the macroblock, the input buffer feeds the edge filter (shown on Figure 14b) for the vertical edge borders filtering.
The result of the vertical edge border pixels is transposed and stored on a macroblock buffer to be used on a second pass through the edge filter. In this second pass, the horizontal edges are filtered and the result is sent to the output. The Line Buffer stores one line of 4x4 blocks of the image to enable filtering between vertically adjacent macroblocks. The Line Buffering is filled as the line of macroblocks passes through the deblocking filter. The edge filter is pipelined in order to achieve high speed and it performs the filtering of an edge in 16 cycles. On the first pipeline stages, Ap, Aq, Alpha, Beta and Delta are calculated. After, different datapaths calculate the outputs for BS={1,2,3} and for BS=4 for the input samples. On the last stage the correct output is selected for any given situation. Pipeline operation enables filtering one macroblock in 256 cycles. Four filtered image samples are produced in parallel.

3.3.5 Video Decoder Hardware Tests and Validation

Video decoder hardware modules were integrated to generate a single system as described before in

![Deblocking Filter Architecture](image)

![Encapsulated Edge Filter](image)
Figure 9. The system validation and integration was performed in different development phases. On a first development phase, each module was validated independently of the entire system. In this strategy, individual module inputs and expected outputs were obtained from a reference software implementation and compared with the hardware outputs. On the second development phase, each module was validated with the use of an external processor (Rosa et al., 2007b) which feeds inputs and checks the outputs both produced with the same reference software, as shown in Figure 15. In this step, modules can be concatenated by the processor gathering data from one module and feeding another module in the sequence. This method has as advantage the possibility of testing the modules at full clock speed for small amounts of data. As drawbacks, continuous operation for large inputs (a few seconds QCIF video for instance) is slow as it demands time to feed inputs and check outputs by software, at the speed of the processor. Also, image display is very slow when it is composed by the same processor.

![Figure 15: Second phase of system validation and integration (Rosa et al., 2007b).](image)

On the third development phase, the modules were integrated without the processor with the use of FIFOs and point-to-point interconnections, enabling the operation of the system in real time, as presented by Bonatto et al. (2010b). Therefore, by using this method it is possible to validate the video decoder operation by simulation. Distinctly from the previous phase, this methodology avoids the need of a simulation model for the processor which is difficult to obtain and its simulation is complex. On this project, this third phase is performed incrementally. It has started with an small version of the video decoder that is capable of decoding only a subset of the standard (including only frames with intraframe prediction on the baseline profile without reference frames storage). The remaining features are being included progressively (i.e. external memory access, motion compensation, deblocking filter, and interlaced video support). Some important issues and design strategies used to perform system integration are also presented by Soares et al. (2011).

3.4 ASIC Design

A commercial video decoder will be used as an ASIC (Application Specific Integrated Circuit) in order to present low cost for a large production volume and low power consumption. The FPGA implementation is a real proof-of-concepts of the desired video decoder architecture, but with different performance, logic area and power consumption of a real ASIC. Nevertheless, the FPGA implementation is used to obtain a better characterization of the prototype. An H.264/AVC version of the video decoder, without the Filter and MC, was synthesized to standard-cells using a TSMC 0.18-μm with 6 metal layers
in CMOS technology (Artisan, 2003). It was also constrained to run at 50 MHz clock frequency keeping the same throughput presented in the FPGA implementation. This clock frequency perfectly meets the required speed to decode HD 720p videos at 30 frames per second, also minimize the dynamic component on the total power consumption of the ASIC implementation.

As result of the ASIC design, a fully verified design layout was produced with 5kB of on-chip SRAM memory and 150k equivalent-gates in a 2.8mm×2.8mm area. The memory blocks are responsible for 19% of the layout area. Figure 16 presents the video decoder layout, in which the memory distribution can be clearly seen over it. Table 3 shows the gate count for each hardware module and for the whole intra-only video decoder. Also, a comparison with other works found in the literature is presented in Table 4.

![Image: Video decoder ASIC layout](image-url)

**Figure 16:** Video decoder ASIC layout (Silva & Bampi, 2010).

<table>
<thead>
<tr>
<th>Hardware Module</th>
<th>Gate Count (k)</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parser</td>
<td>59.3</td>
<td>39.6%</td>
</tr>
<tr>
<td>ITIQ</td>
<td>50.5</td>
<td>33.7%</td>
</tr>
<tr>
<td>Intra</td>
<td>28.6</td>
<td>19.2%</td>
</tr>
<tr>
<td>FIFOs</td>
<td>11.6</td>
<td>7.2%</td>
</tr>
<tr>
<td>Adder</td>
<td>0.5</td>
<td>0.4%</td>
</tr>
<tr>
<td>Top</td>
<td>150.5</td>
<td>100.0%</td>
</tr>
</tbody>
</table>

**Table 3:** Gate count of the Intra-only decoder ASIC design.
4 Digital Television Set-Top Box Hardware Architecture Design

In this section, digital television set-top box hardware architecture is presented and discussed. A set-top box is composed by several processing units, treating data at many points of the data flow and in different levels. As illustrated in Figure 17, the system can be subdivided into user interface and application unit, analog/digital data input unit, analog/digital data output unit and data signal decoding unit. In the following sections we analyze this system in more detail.

![Diagram of SBTVD Set-Top Box Hardware Architecture Design](image)

**Figure 17:** The proposed architecture for the SBTVD Set-Top box is shown in this figure by its main units. It is a complex mixed-signal processing system, designed to be integrated into a single silicon chip.
4.1 Digital Television Set-top Box Requirements

In a typical set-top-box architecture a RF front-end provides a transport stream that contains video, audio and control information for a selected RF channel. An MPEG Transport Stream (TS) (ISO/IEC, 2000) can carry several audio and video programs and other information like the program time table. The selection of the correct (audio, video, closed captions) streams is performed by the demux, according to the user selection and to the program tables contained in the TS. The demux provides the separated audio, video and closed captions elementary streams to the appropriate decoders. The video decoder will produce image frames that must be stored and processed prior to being sent to the video composition block. Modern video decoders like H.264 will decode video frames in an order that is different from the exhibition order, so video frame reordering must be performed after decoding and prior to exhibition (Wiegand et al., 2003).

The number of decoded video frames per second (FPS) must also be matched to the display exhibition rate. For low frame rate videos, common in portable devices, it may be necessary to send the same video frame to the display several times. The video frame repetition and reordering must be performed after the decoder and prior to being sent to the display.

Besides decoded video, other image layers must be overlaid in the display. Closed captions must also be decoded, rendered according to the decoded video resolution and overlaid to the decoded video. Also, CPU generated graphics like menus must also be overlaid to the decoded video. The combination of several video sources is carried out by the video composition and scaler block. A video scaler is needed since the video resolution will generally not match the display resolution, or video miniatures may be required by the user interface. Similar to the video content, audio must also be combined with CPU generated audio prior to being sent to the audio controller.

Decoding of different content is independent and takes different time in order to complete. Therefore it is necessary to establish some sort of synchronization between the decoded audio, video and closed captions. A program clock reference is present in the transport stream and must be decoded in order to synchronize the clock of the decoder with the clock of the encoder. The recovered clock can be used to generate a system clock base needed to synchronize video, audio and closed captions. The user interface may be implemented in a high level language on a CPU, being able to access peripherals like the remote control interface and configuration registers of individual hardware components. These requirements are addressed in the following sections.

4.1.1 RF Frontend

In order to obtain a TS a typical set-top-box will need a RF frontend suitable for the target terrestrial digital television system. As presented before in Table 1, there are several terrestrial digital television systems and the RF frontend will be targeted to a specific system in order to maximize performance. The RF frontend needs to tune to a specific frequency channel and provide a baseband modulated signal to a demodulator. The channel selection and RF receiver setup must be performed by the set-top-box CPU according to the user selection and to the TV broadcast stations in the range of the receiver. In order to control the RF frontend, a serial protocol like i2c may be used by the set-top box CPU. After being correctly configured, the demodulator will then perform the required demodulation operations according to the specific digital television standard used. Besides demodulation, error correction will also be
performed, since errors are likely to occur due to varying signal strength, for example. The error corrected data stream output by the demodulator is a MPEG-2 transport stream which carries the program data from the DT broadcaster. The transport stream data must be routed to the demultiplexer in order to obtain the separated video, audio and closed captions streams.

### 4.1.2 Transport Stream Processing

The MPEG-2 transport stream uses a 188 bytes length packet stream with a header and payload (ISO/IEC, 2000). In the header there is a transport error indicator that will be set if the RF frontend and demodulators were not able to correct errors in the received TS. There is also a continuity counter that can be used to check if packets of the same PID are being received in the correct order. The demux is responsible for analyzing the TS and output the elementary streams to the corresponding decoders. In order to perform this operation, the demux must first receive the program tables, which will identify the number of programs being sent in the TS, and associate the correct video, audio and closed captions streams to each program. Besides processing the streams the demux must also analyze the program clock reference, needed to synchronize the streams, and other information like the electronic program guide.

### 4.1.3 Video, Audio and Data Decoders

Depending on the digital television standard being used, the correct decoders must be implemented in the set-top box. For ISDB-T International (SBTVD), an H.264/AVC video decoder and an MPEG4-AAC audio decoder are needed. The video decoder will perform rendering of the video frames and output data to a video memory buffer. The video memory must be able to store a number of high-definition video frames. Furthermore, during the decoding process, the decoder will need access to previously decoded images as reference images. This way, a somewhat large memory area is needed, with fast access to support full-HD image resolutions and display frame rates. The audio decoder performs audio decoding, which does not require previously stored frames or access to previously decoded data. The audio stream must be decoded as needed in order to maintain synchronization with the video stream, and the output streams can be stored in an output audio memory for post processing and mixing. The closed captions decoder must be able to retrieve the text information and to render an image with the received text that must be overlaid to the video frame. The text information can be retrieved and stored in a buffer, with the rendering and overlay being controlled by a dedicated module in the video composition and scaling stage.

### 4.1.4 Data Buffers, Synchronization and Post Decoding Processing

In order to allow synchronization between video, audio and closed captions, it is necessary to separate the output of the decoders from the data being sent to the output devices. This is needed since the decoders do not operate always at the same time an have different delays. A memory buffer provides a way to control the output of the decoded data streams and to compensate for decoding delays or video frame reordering. Post decoding processing involves all processing not done by the decoders, like video scaling, video overlaying, video layer selection, closed captions rendering and overlaying, audio mixing, etc.

### 4.1.5 System Control and User interface

The basic system control are the necessary functions that need to be performed for the basic operation of the set-top-box, like RF frontend setup, setup of decoders and post processing blocks, display and audio
adjustments and initializations. Basic user interface systems will allow the display of set-top-box information overlaid to the decoded video, like menus. More sophisticated applications may be required on a digital television. In the SBTVD system, Ginga (Baum & Soares, 2012) is a middleware specification that allows the deployment of applications based on Java or NCL (Nested Context Language).

### 4.1.6 Multichannel Memory Controller and Decoded Picture Buffer

The set-top box architecture includes a multichannel memory controller (MMC) module to implements a Decoded Picture Buffer (DPB) for the H.264/AVC decoder and for the video post decoding processing modules. A DPB is necessary to store decoded pictures for reference, by the motion compensation process, and for enabling output reordering and output delay necessary in video output synchronization. The MMC needs to interface with a large size external SDRAM memory (DDR, DDR2 or DDR3), or internal SRAM. Since SRAMs are costly and have large power consumption, video decoding systems prefer external SDRAM memories. Video processing systems require an efficient memory hierarchy design to achieve real-time performance for the decoding of high definition video sequences. Both the decoded video and the on-screen display data are stored in the same shared memory to be composed by the video output module. For this large size memory, an off-chip DRAM (Dynamic Random Access Memory) is used and the memory accesses are directed to a single off-chip memory interface. A memory hierarchy is necessary to optimize the interface between system and external memory reducing data latency, to generate memory addresses and to isolate different clock domains. A four-level memory hierarchy was designed to manage the decoded video with granularity of macroblock of pixels with low latency (Bonatto et al., 2010a).

Data latency can affect the system performance when several modules are sharing the same memory. In video decoding systems, reducing latency is a critical design task to reach real-time decoding. Latency depends on how data is transferred between the channels and memory and also depends on the data transfer granularity and frequency. Because data accesses to double data rate SDRAM memories are burst-oriented (JEDEC, 2009), dedicated data buffers are used in each interface to the external memory to reduce latency generated by memory row and column activation. The hardware modules accessing external memory have different access behaviors and bandwidth requirements, thus the memory controller has to balance and optimize the data access sequences requested by different modules.

### 4.2 Example Architecture Design for a DT Set-top box

In this section the set-top box implementation example presented in Figure 18 is discussed. A commercial RF frontend is used in order to provide TS data. Since this specific frontend must be programmed through an i2c interface, an i2c controller is needed and should be connected to the set top box CPU. In this example a Leon3 processor (Aeroflex, 2012a) is used as the set top box CPU. The processor interfaces with the main memory and with other peripherals through a bus compatible with the Amba 2.0 specification from ARM and several peripherals are available through an IP core library (Aeroflex, 2012b). In order to off-load the CPU, data and CPU demanding tasks should be implemented on hardware.

Following this approach, hardware implementations of the demux and of the decoders are presented in Figure 18. Since the demux must pass program information data to the CPU and receive the
desired program selection from the user, the demux must also be interfaced to the Amba bus. Several functional units must have access to memory. In order to save costs it is necessary to share access to external memory, and this can be accomplished by using a multichannel memory controller. This way all hardware modules and CPU can share the same physical external memory.

External DDR2/DDR3 memory modules have large data capacity and are fast, being ideal for such application. The memory controller will time share the access to the external memory, so accesses to external memory may take longer to complete. This also means that all modules accessing external memory must be designed in order to accept such delays. The video decoder must be able to store the decoded video frames and, in order to perform decoding, it also needs access to previously decoded frames. As a result, different memory interfaces must be implemented for the video decoder. The decoded video frames are then reordered in the video channel interface by the decoded picture buffer, which is read by the post decoding processing block.

Figure 18: Digital Television Set Top Box implementation example.

The post decoding processing block is responsible for synchronizing the output streams and for performing the necessary operations in order to present audio and video data adequately to the display and audio controllers. Video composition and scaling are required operations and its hardware implementation is discussed in more detail in (Negreiros et al., 2012). An implementation of the AAC audio decoder is discussed in (Renner, 2011). CPU-generated content is stored in external memory and accessed by the post decoding processing block. The CPU is able to control the video and audio composition and selection of the post decoding processing block by accessing configuration registers of the post processing blocks. By off-loading the CPU from decoding tasks the architecture enables the CPU to run a complete operating system, providing flexibility to implement more sophisticated user interfaces and providing support for middleware like Ginga, besides enabling system interfaces like Ethernet and USB.
5 Conclusions

Multimedia objects like images, music and videos use large amounts of information when digitized. A big effort is been made by the scientific community to compress the information of multimedia objects exploiting redundancies and psycho-acoustic and psycho-visual characteristics. These objects are used to register or transmit real life scenarios and are meant to be exchanged between people from different countries and cultures and they are supposed to be coded according to standards in such a way that devices designed all over the world can correctly recover. This applies to television signal and in the case of this paper, to Digital Television signal.

In this paper we have presented the fundamentals of video decoding in the H.264/AVC Standard and its architectural implementation in a digital system. Also, it is presented an architectural description of a digital television set-top box implementation which has a H.264 video decoder. As a result of this work it has been implemented the first Brazilian H.264/AVC video decoder prototyped on FPGA integrated to a digital television set-top box SoC. The video decoder implemented prototyped in FPGA is able do decode high definition video sequences supporting I and P frames. Its ASIC implementation was also designed and validated. However, it has not been sent to foundry yet. It is composed of a set of modules which perfectly enables it to decode videos in different video resolutions, since QVGA to HD.

The complete Set-top Box is also prototyped in FPGA and currently is being validated. A complete SoC is a very large digital system integrating the audio and video decoders with a CPU and peripherals that manages the operation of the Set-top Box, the Operating System and the middleware and the user interface. In our FPGA prototyping design we are using the open Leon3 (Aeroflex, 2012a) as the CPU. The optimization of the memory usage, an external DD2 memory, was very challenging as the many processes use the external memory and a Multichannel Memory interface IP was specially designed.

Even though the many blocks run correctly alone or connected to its neighbors, the integration of all the hardware and software pieces will need some work. The choice of the target technological node is being discussed. The designs done so far were designed targeting the available foundry at the moment. For a complete SoC we will have more constraints like specific financial support.

For this project we are thankful to Capes, CNPq, CTIC and Finep, Brazilian research financing agencies for projects RH-TVD, Namitec, SoC-SBTVD and Rede H.264 for the students' scholarships.

References


