# MAC_MPEG2_AV: A Low Energy Implementation of an Audio/Video Decoder

Adam B. Kinsman and Nicola Nicolici, McMaster University, Ontario, Canada

## 1 Introduction

The Moving Picture Experts Group (MPEG) is a working group of ISO/IEC who oversee development of international audio/video compression/decompression standards. Their standard for digital television, known as ISO/IEC 13818 or MPEG-2 [1] was approved in November 1994, and has become widespread as a storage/transmission format for digital video. While the conception of MPEG-2 was based on digital television, the ever increasing demand for personal media devices as motivated by the digital lifestyle movement is making MPEG-2 attractive for use in personal playback devices. In such an environment, energy efficiency will be of paramount importance to enable extended battery lifetimes. As a result, the goal of this project has been to provide a low energy implementation of the MPEG-2 video decoding process, as well as the corresponding layer 2 audio decoding process. We have focused on main-level, main profile coding (4:2:0) at 720x480@29.97 fps (NTSC) in accordance with the popularity of this configuration.

## 2 MPEG-2 Algorithm

In this section, we provide a brief summary of the MPEG-2 compressed data flow as it applies to our specific implementation. Full details can be obtained from the standard [1], part 2.

During compression, a stream of video is sampled into *frames*. Each frame which is compressed can either be coded as-is (*I-frames*), or predictively (*P-* and *B-frames*) by using a temporally adjacent frame as a reference and encoding only the difference from the reference frame. In prediction, only I- or P-frames are used as reference frames, and thus are referred to as *anchor frames*. P-frames only use the previous anchor frame as a reference (forward prediction) while B-frames may use as a reference either the previous (forward prediction) or subsequent (backward prediction) anchor frame. To enable this scenario, the frames are transmitted out of order in the bitstream (each anchor frame before the first place it is used as a reference).

The frames themselves are divided into luminance (Y) and chrominance (Cb, Cr) *planes*. Chrominance planes are downsampled and the data for a 16x16 array of full colour pixels is organized into a *macroblock* (in 4:2:0 this requires 6 8x8 blocks of pixels: 4 Y blocks, 1 Cb block and 1 Cr block). The above mentioned prediction actually occurs at the macroblock level, and *motion vectors* accompany each macroblock to indicate the location of the reference macroblock. The picture data (original Y, Cb, Cr values for I macroblocks, difference Y, Cb, Cr values for P/B macroblocks) undergoes Discrete Cosine Transform (DCT), quantization, run/level encoding and Huffman encoding before being packaged into the bitstream. Furthermore, additional control in the bitstream allows some macroblocks to be *skipped*, meaning they are entirely predicted from the corresponding reference macroblocks (no difference data is required).

Audio encoding is much more straightforward, the sampled audio stream is broken into frames of 1152 samples, and each frame is broken into blocks which are divided into sub-bands, then quantized and packaged, allocating as many bits as needed for each sub-band. Full details may be obtained from part 3 of [1].

## 3 MPEG-2 Architecture

Figure 1 shows the architecture and interface for our energy conscious implementation of the MPEG-2 audio/video decoding flow. Energy efficiency has been addressed by both reducing the amount of hardware required, and by reducing rate at which that hardware is clocked. Section 4 elaborates on the required clock rate and hardware resources.

### 3.1 Video Decoding Pipe

The main core of the video decoding circuitry exists in the $\underline{Sequence\_Decode.v}$ section of Figure 1 which operates exclusively at 54 MHz. The audio pipe is contained in $\underline{MP2\_Decode\_16.v}$ and operates at 27 MHz. The remaining circuitry provides support for validating the design on Xilinx's Virtex-2 Multimedia Development Board [2]. During playback, an MPEG-2 file is streamed to the board via the ethernet port and buffered in external RAM. The audio and video streams are extracted from the system level bitstream (see part 1 of [1]) and buffered externally. Each decoder requests data from these buffers using shift control signals (byte shift / bit shift).

In the video pipe, the first major breakpoint occurs where data is buffered between slice decoding and inverse quantization, mainly because this is the barrier between variable demand for data and uniform demand for data. This buffer is a dual-port RAM, which is monitored, allowing slice decoding to proceed only while there is room in the buffer. Concurrently, decoded info processing extracts control info and data from the decoded info buffer and directs processing, i.e. starts prediction and quantization as necessary. For P/B macroblocks a prediction is formed and for non-skipped macroblocks the run/level data are de-quantized to obtain Inverse Discrete Cosine Transform (IDCT) coefficients.

IDCT, undoubtedly the bottleneck of the decoding process, transforms the coefficients (using fixed point arithmetic to conserve energy over floating point), producing 1 block of 64 samples of IDCT data per 212 clock cycles, or 1 block every $3.93\mu s$ at 54 MHz. Given that a 720x480 image contains $720/16 \times 480/16 \times 6 = 8100$ blocks (at 4:2:0) and must be completed in 33.37 ms (in accordance with 29.97 fps), the IDCT is busy roughly 95% of the duration of the frame, leaving 5% for setting up the decoding pipe (e.g. slice decoding and dequantization).

The finished IDCT data for each macroblock (in 4:2:0 = 4 Y blocks, 1 Cb block, 1 Cr block) is mixed with the corresponding prediction data to form the completed macroblock. Since, as mentioned above, frames are transmitted out of display order in the bitstream to enable backward prediction, the $\underline{Framestore\_Management.v}$ unit takes care of presenting the frames to the display in the proper order, as well as managing the access to the physical memories and updating the framestores when a new anchor frame is decoded (due to physical resource limitations, the audio and video bitstreams must be buffered in the framestores and this unit also arbitrates that access). This choice of the second major breakpoint is mainly due to the fact that anchor frames must be buffered at this point anyway to be used in prediction. Finally, the $\underline{Backend.v}$ unit performs real-time upsampling and colourspace conversion on the 40 MHz clock domain, delivering data to the SVGA port.

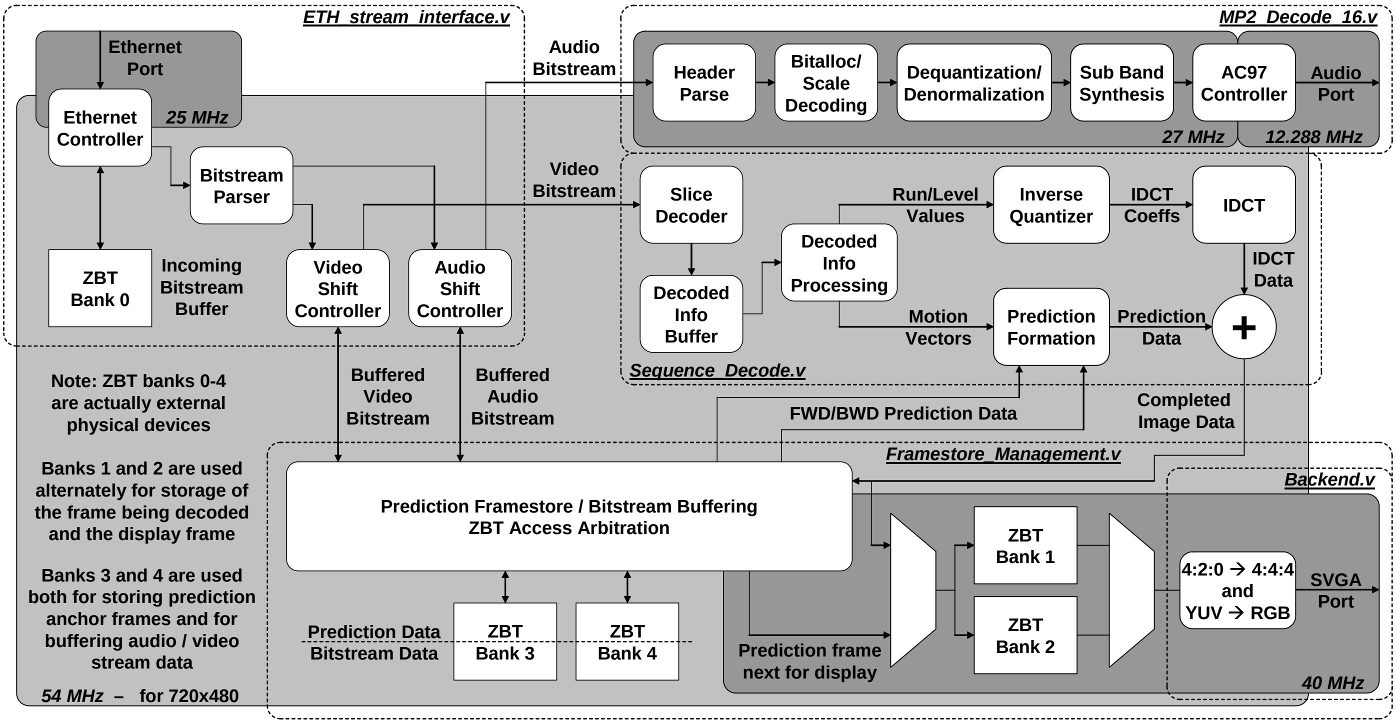

**Figure 1. MPEG-2 Audio/Video Decoding Pipe and Support Circuitry Overview.**

### 3.2 Audio Decoding Pipe

In the audio decoder, header info, bit allocations and scale factors are provided for a set of blocks, then sub-band samples are processed and synthesized. Fixed point arithmetic (16-bit) is utilized to reduce the resource requirements over the case for floating point, for energy's sake. Bit allocations and scale factors must be buffered (since they are used over multiple blocks) and to be energy efficient, the first breakpoint consists of only a single block of 32 samples being buffered before sub-band synthesis.

Where IDCT is the bottleneck during video decoding, sub-band synthesis and windowing dominate the computational load for audio decoding, requiring $(64 + 16) \times 2 = 160$ clock cycles per sample (64-sub-band, 16-window, 2-channels).

The second audio breakpoint occurs within sub-band synthesis where 1024 samples of each channel are unavoidably buffered for the windowing filter. After this the completed samples are passed through the AC97 controller to the audio port.

## 4 Implementation Results

As mentioned previously, both resource requirements and clock frequency contribute to energy consumption and as a result, our approach to providing an energy efficient implementation has been to reduce the resources as much as possible, and to clock the required resources at as low a frequency as possible.

The overall implementation in the Virtex-2 device on the Xilinx Multimedia Board [2] (as depicted in Figure 1) consists of 3,337 FFs and 10,615 LUTs, with 28 BRAMs and 3 MULT18X18s. Regarding clocks, 6 GCLK resources driven by 4 DCMs were required. Peripherals on the board which were utilized were the Audio Codec, the SVGA DAC and the Ethernet PHY, as well as all 5 external ZBT memory banks. The figures reported above include all support circuitry for the complete implementation of Figure 1.

The core of the video decoder (Figure 1 - *Sequence_Decode.v*) which processes the video bitstream and produces a 4:2:0 YCbCr picture required 980 FFs, 4,683 LUTs, 13 BRAMs and 1 MULT18x18, all operating at 54 MHz. The 4:2:0 upsampling and colourspace conversion circuitry (*Backend.v*) required 349 FFs, 2,059 LUTs, 2 BRAMs and 1 MULT18X18, all operating at 40 MHz. The framestore manager (*Framestore_Management.v*), which contains the upsampling circuitry reported above required 330 FFs and 653 LUTs on top of the *Backend.v* resources, and the unit straddles the 54 MHz / 40 MHz clock domains. The audio decoder (*MP2_Decode_16.v*) including the AC97 controller required 556 FFs, 1568 LUTs, 4 BRAMs and 1 MULT18X18 and operates at 27 MHz, except for the AC97 controller which operates at 12.288 MHz. Note that the 27 MHz clock for audio is used for convenience, but is $\approx 3\times$ faster than required. Finally the ethernet unit which brings the data onto the chip uses 1006 FFs, 2110 LUTs, and 9 BRAMs, and operates mainly on the 25 MHz clock domain, but does cross the 25 MHz / 54 MHz clock domain border.

## 5 Acknowledgements

The authors acknowledge Phillip Kinsman for his contribution of the 4:2:0 to 4:4:4 upsampling and display circuitry as contained in the *Backend.v* section of Figure 1. In addition, Jason Thong implemented the adder-tree multipliers for the fast-IDCT, and the combinational colourspace conversion circuitry. Although code developed by Matthew Carrier and Mark Jobes was not in the final release, their implementation efforts and input to discussions have been appreciated. Finally, the authors gratefully acknowledge the Canadian Microelectronics Corporation (CMC) for the provision of FPGA prototyping platforms and CAD design software.

## References

[1] ISO/IEC. *Information technology Generic coding of moving pictures and associated audio information: parts 1-3*, second edition.

[2] Xilinx Incorporated. Xilinx Multimedia Board. http://www.xilinx.com/products/boards/multimedia/, March 2007.