FPGA-based MP3 Player

Project Proposal

Student: Lalu Luka

Advisors:
Dr. Yufeng Lu
Dr. In Soo Ahn

Bradley University

December 10, 2009

Introduction

Digital design using a Field Programmable Gate Array (FPGA) is a rapidly evolving field. A complete embedded system can be built and programmed into a single FPGA chip for digital signal processing applications.

The goal of this project is to design a MPEG Layer III (MP3) player using a FPGA board. The FPGA will read MP3 source files, decode them into a 16-bit Pulse Code Modulated (PCM) output, and play the audio files through an external speaker.

Project Goals

- Gain an in-depth understanding of hardware/software co-design using FPGAs.
- Understand the specifications set by the ISO/IEC 11172-3 standard for encoding and decoding MP3 files.
- Build a FPGA-based MP3 system, which implements the MP3 decoding algorithm using VHDL and C language on the Embedded Development Kit (EDK) software platform.
- Design a user friendly interface that
    - Enables the user to scan through the MP3 file list, and then select, play, pause, and/or stop the song.
    - Displays related information on the LCD such as song title and author(s).
    - Advanced features, such as volume control, rewind and forward modes will be added to the system, if time permits.

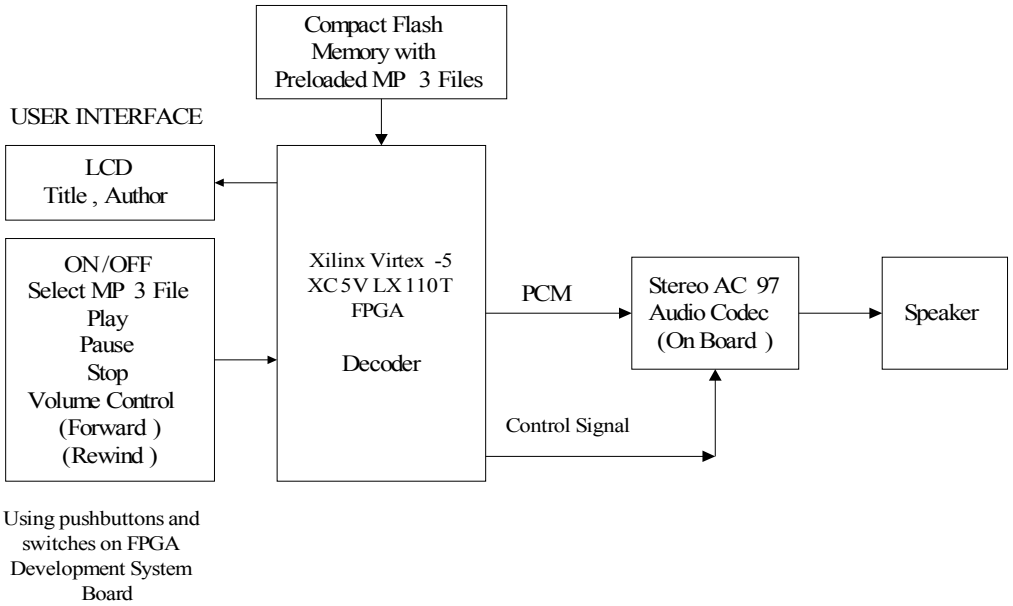

Figure 1 High Level System Block Diagram

The system includes:

- **User Interface**
  The user interface provides the inputs to control the MP3 player, such as scanning, selecting, playing, pausing, and stopping the MP3 files. It will also allow to output related information (i.e, title and author of the song) on the LCD.

- **Compact Flash Memory Card**
  The compact flash memory (CFM) card supplies the preloaded MP3 files for the MP3 decoder system in the FPGA. MP3 files are loaded onto to the CFM using a PC and memory card reader.

- **Onboard Stereo Audio AC97 Codec**
  The AC97 codec (i.e., AD1981B) is used to convert the PCM format signal from the MP3 decoder into an audio signal, which is fed into an external speaker through an audio jack.

- **MP3 Decoder**
  A MP3 decoder runs on the Xilinx Virtex-5 XC5V LX110T FPGA that will decode the selected MP3 stream with the sampling frequency specified in the MP3 header. A typical sampling frequency is 44.1 kHz.

Decoding Process Overview:

The decoding process is shown in Figure 2. It includes the following stages: initial reading, Huffman decoding, re-quantization and reordering, stereo decoding, alias reduction, inverse modified discrete cosine transform (IMDCT), and synthesis polyphase filter bank [1-7].

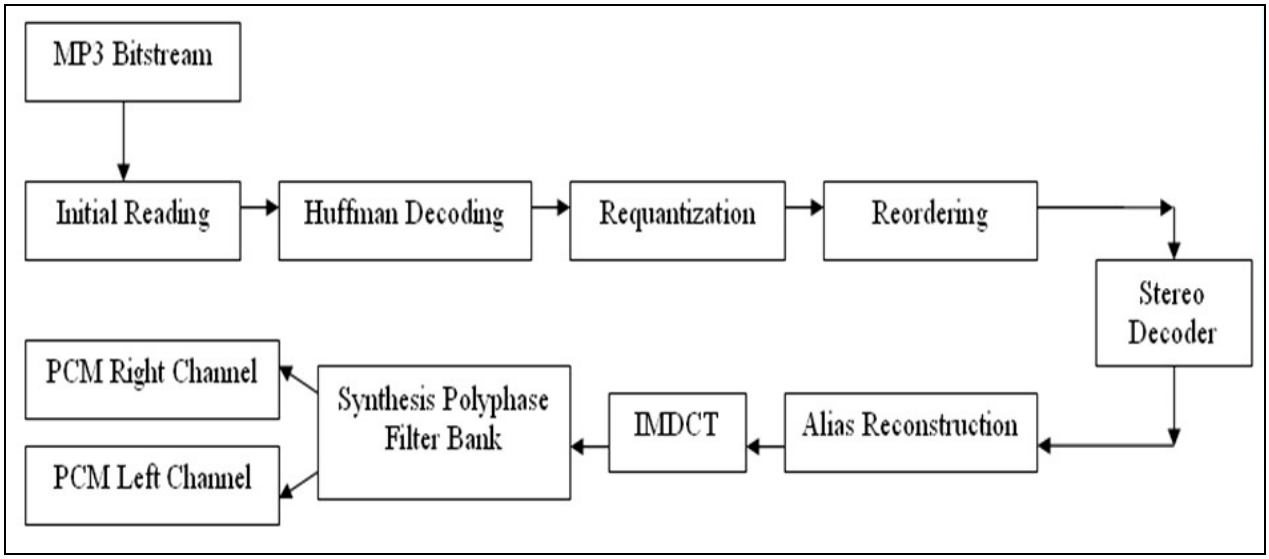

Figure 2 High Level Diagram of Decoding Process

1. **Initial reading**
    - The incoming data stream is split up into individual frames. The header section of each frame is analyzed to obtain parameters used in the encoding process (i.e. bit rate and sampling frequency).

2. **Huffman decoding.**
    - The Huffman algorithm is used for lossless data compression.
    - The basic idea of the technique is to assign shorter binary codes to more frequent samples and longer codes to less frequent samples.
    - The decoding is based on tables that are used to map the Huffman binary codes to the original samples.

3. **Re-quantization and re-ordering**
    - The Huffman decoder output is scaled up for different frequency spectrum ranges and sorted by increasing frequency ranges.

4. **Stereo decoder**
    - The output from re-ordering block split into two identical outputs for the right and left channels.

5. **Alias reduction block**
    - In the MP3 encoding process, anti-aliasing filters have been added to the signal. To obtain a correct reconstruction of the signal in the MP3 decoding process, special designed butterfly calculations are applied to each frequency spectrum range to reconstruct the aliasing artifacts.

6. **Inverse Modified Discrete Cosine Transform (IMDCT)**
    - A counterpart of the MDCT in the encoding stage of MP3 file.
    - DCT is an energy-compaction transformation for real-number signals.

7. **Synthesis polyphase filter bank**
    - Compared with analysis polyphase filter bank, synthesis polyphase filter bank is used to combine the signals from different frequency spectrum ranges.
    - The output of the synthesis polyphase filter bank is in PCM format.

System functional requirements and performance specifications:

- **Input MP3 bitstream requirements**
  The MP3 player will decode MP3 inputs with various bit rates (from 128 kbps to 320 kbps) and different sampling frequencies (32 kHz, 44.1 kHz or 48 kHz)

- **Decoding speed**
  The ultimate objective of decoding speed is to process MP3 files in real-time. The execution time of the MP3 decoding will be profiled and measured. If the real-time specification can not be met, further optimization will be needed.

- **LCD display**
  The LCD will list the MP3 information in real-time.

Preliminary progress:

The first few weeks have been dedicated to becoming familiar with the Embedded Development Kit Platform Studio software. Xilinx tutorials and board demonstration projects have been studied. In addition, peripheral setup, compilation, and debugging procedures have been practiced extensively.

C programs have been written to read from a compact flash memory (CFM). First, the CFM is preloaded with .txt or .dat files. Then, the C program running on the FPGA fetches the file (.txt or .dat) and outputs the contents on a hyperterminal. It proves that CFM is accessible using C programming. In the actual implementation of the project, MP3 files will be preloaded instead.

Patent search:
Reference codes for existing MP3 decoders were obtained. All software provided requires no license and users are free to implement or modify them (see [8] and [9]).

Equipment list:

Table 1. Equipment list

Item	***Description***
Xilinx UPV5-LX110T FPGA	FPGA used to implement decoding process
AD1981B JSTZ	Onboard AC97 Stereo Codec Chip used to convert decoder digital PCM outputs to analog audio sound waves.
1 GB Compact Flash Memory (CFM) and an external memory card reader	The CFM holds preloaded Mp3 files to be accessed and decoded by FPGA system. Eventually decoding process software will be run from the CFM, so that an external PC is not required.
Headphones or external speaker	Headphones will be connected through the AC97 audio jack (driven by the audio codec's internal 50-mW amplifier).

Preliminary schedule:
The schedule for the Spring 2010 semester is given in Table 2, below. A significant amount of time will be used to integrate the MP3 system.

Table 2. Spring 2010 schedule

Week #	Dates	Activity
1-2	01/24-02/06	Stereo AC97 interface.
3-4	02/07-02/20	LCD and user interface setup.
5-8	02/21-03/20	MP3 decoding system (Huffman decoding, alias reduction, IMDCT, synthesis filter) testing and debugging
9-10	03/21-04/03	System performance evaluation, optimization
11-14	04/04-05/01	System optimization, debugging, and documentation.

References

[1] A. Abdel-Gawad, "A *full hardware implementation for an MP3 decoder chip using VHDL*", Project Report, University of California at Santa Barbara, December 2008.

[2] M. Botteck, H. Blume, J. von Livonius, M. Neuenhahn, and T. Noll, "*Programmable architectures for realtime music decompression*", Proceedings of Parallel Computing: Architectures, Algorithms, and Applications Conference, vol. 38, pp. 777-784, September 2007.

[3] K. Brandenburg, and H. Popp, "*An introduction to MPEG layer-3*", European Broadcasting Union Technical Review, pp. 1-15, June 2000.

[4] P. Chandraiah, and R. D¨omer , "*Specification and design of a MP3 audio decoder*", Technical Report CECS-05-04 , University of California at Irvine, pp. 1-83, May 2005.

[5] International Standard ISO/IEC 11172-3. "*Information technology - Coding of moving pictures and associated audio for digital storage media at up to about I,5 Mbit/s – Part3: Audio,"* January 1999.

[6] W. Jiang, S. Polisetty, and X. Li, "Huffman decoder", Project Report, University of Tennessee at Knoxville, May 2003.

[7] G. Liaskos, "MP3 File Structure", Internet Resource : http://www.multiweb.cz/twoinches/MP3inside.htm#MP3FileStructure

[8] R. Leslie, "MAD: MPEG Audio Decoder" Internet Resource: http://www.mars.org/home/rob/proj/mpeg/

[9] M. Hipp and O. Fromme. "MPG123" Internet Resource: http://www-ti.informatik.uni-tuebingen.de/~hippm/mpg123.html