EngrECE 298
System-on-Chip Description and Modeling
Spring 2004

Assignment 1

Posted: April 23, 2004 (week 3)
Due: April 30, 2004 (week 4)

Task: Get familiar with MPEG-Audio

Instructions:

The goal of this first assignment is to make yourself familiar with the application that we will be using for the following assignments, namely MPEG Audio. MPEG Audio is a coding standard for audio files. It is one of the results of the work of the Moving Picture Experts Group (MPEG).

Task 1: Plenty of information is available online about MPEG at http://www.mpeg.org/MPEG/index.html. Make yourself familiar with MPEG Audio by using information from this web page.

Task 2: A free implementation of the MPEG Audio layers 1, 2 and 3 is available online at http://www.mpg123.de/. Download the latest version of the mpg123 software (http://www.mpg123.de/mpg123/mpg123-0.59r.tar.gz) and install it on your Unix account.

Task 3: Try the software by encoding and decoding examples of audio files with it. As examples, you may use sound files that you own or free music that can be downloaded from the web. Make sure, however, to obey all copyright restrictions! For the following assignments, we will need a single audio file (both, in encoded and in decoded format) that can be used for building a test bench for the codec that we will develop later.

Task 4: Make yourself familiar with the implementation of the software. Browse through the different C source files and try getting an overview on its complexity, composition, and structure. Note that it is *not* required that you understand each line of code, but you should reach an understanding of the basic software architecture.

In particular, try answering the following questions:

- How many source files are there? What is their dependency?
- How many lines of code does the code consist of?
- How many lines of code are used as environment *around* the actual encoder and decoder (those will later become part of the test bench)?

- Which parts of the code (and how many lines of code) are used for the encoder? Which lines and how many for the decoder?
- How many functions are there?
- What is the calling tree of the functions?
- What kind of operations are used in the algorithms?
- Are there any special things that need to be considered when we aim to create an implementation of the MPEG audio algorithms as an embedded system later?

--
Rainer Doemer (ET 444C, x4-9007, doemer@uci.edu)