International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 4, Issue 7 (November 2012), PP. 21-24

FLAC Decoder using ARM920T using S3C2440

J. L. DivyaShivani[1,] M. Madan Gopal [2]
[1]*M.Tech (Embedded Systems) Student,* [2]*Assoc.Professor Aurora's Technological & Research Institute Uppal, Hyderabad, INDIA*

Abstract: *In this paper, an embedded FLAC decoder system was designed, and the embedded development platform of ARM920T was built for the design. Furthermore, the IIS bus of S3C2440 in Linux which were used in designing the decoder system. Results show that the FLAC format sound can play well in the decoder system. The decoding solution can be applied to many high-end audio devices.*

With the development of multimedia technology, as well as the people's requirements to higher sound quality, the Lossy compression coding audio format such as MP3 cannot satisfy many music lovers. Therefore, many R & D staffs have research on how to develop Lossless Audio Decoding systems based on embedded devices with lower price and better sound quality. FLAC stands for Free Lossless Audio Codec, an audio format similar to MP3, but lossless, meaning that audio is compressed in FLAC without any loss in quality. This is similar to how Zip works, except with FLAC you will get much better compression because it is designed specifically for audio, and you can play back compressed FLAC files in your favorite player (or your car or home stereo) just like you would an MP3 file. FLAC stands out as the fastest and most widely supported lossless audio codec, and the only one that at once is non-proprietary, is unencumbered by patents, has an open-source reference implementation, has a well-documented format and API, and has several other independent implementations. It can be predicted that FLAC format will be one of the best promising audio formats, which are expected to replace lossy compression such as MP3 encoding audio and will be the mainstream in the field of multimedia. Up to recent days, there is no hardware decoder chip with FLAC support in chip market. Therefore, the paper made a solution for designing a Free Lossless Audio Decoding system under the Embedded Development Platform of ARM9.

Implementation of FLAC decoder done on ARM9 embedded platform using MINI2440 development board and ROCKBOX firmware and open source player.

I. INTRODUCTION

FLAC stands for Free Lossless Audio Codec, an audio format similar to MP3, but lossless, meaning that audio is compressed in FLAC without any loss in quality. This is similar to how Zip works, except with FLAC you will get much better compression because it is designed specifically for audio.

FLAC stands out as the fastest and most widely supported lossless audio codec, and the only one that at once is non-proprietary, is unencumbered by patents, has an open-source reference implementation, has a well-documented format and API, and has several other independent implementations.

II. FEATURES OF FLAC

a) Lossless: The encoding of audio (PCM) data incurs no loss of information, and the decoded audio is bit-for-bit identical to what went into the encoder. Each frame contains a 16-bit CRC of the frame data for detecting transmission errors.
b) Fast: FLAC is asymmetric in favor of decode speed. Decoding requires only integer arithmetic, and is much less compute-intensive than for most perceptual codecs.
c) Hardware support: FLAC is supported by many consumer electronic devices, from portable players, to home stereo equipment, to car stereo.
d) Flexible metadata: FLAC's metadata blocks can be defined and implemented in future versions of FLAC without breaking older streams or decoders.
e) Seekable: FLAC supports fast sample-accurate seeking. Not only is this useful for playback, it makes FLAC files suitable for use in editing applications.
f) Streamable: FLAC uses sync codes and CRCs (similar to MPEG and other formats), which, along with framing, allow decoders to pick up in the middle of a stream with a minimum of delay.
g) Error resistant: Because of FLAC's framing, stream errors limit the damage to the frame in which the error occurred, typically a small fraction of a second worth of data. Contrast this with some other lossless codecs, in which a single error destroys the remainder of the stream.

III. FORMAT

The basic structure of a FLAC stream is:
a) The four byte string "Flac"
b) The STREAMINFO metadata block
c) Zero or more other metadata blocks
d) One or more audio frames

The first four bytes are to identify the FLAC stream. The metadata that follows contains all the information about the stream except for the audio data itself. After the metadata comes the encoded audio data.
"fLaC", the FLAC stream marker in ASCII, meaning byte 0 of the stream is 0x66, followed by 0x4C 0x61 0x43.
METADATA_BLOCK: This is the mandatory STREAMINFO metadata block that has the basic properties of the stream.
FRAME: Frame of Audio data.

STREAM HEADER

Flac	Metadata block (Mandatory)	Metadata block (Optional)	Frame (One or more audio frames)

META DATA BLOCK

Metadata Block Header	Metadata Block Data

MATADATA BLOCK HEADER gives information of block type.

METADATA BLOCK HEADER

1bit	Block Type	length of metadata to follow excluding Header of metadata block

METADATA BLOCK DATA gives information of minimum block size, minimum frame size, max block size, max block size, sample rate, no of channels, bits per sample and total samples in a stream.

FRAME

Frame Header	sub frame <? >	frame footer

Frame Footer contain information of CRC-16, back to and including the frame header sync code.

SUBFRAME

Sub frame header	subframe constant (or) subframe fixed

SUBFRAME Header give information of Subframetype(CONSTANT, VERBATIM, FIXED, LPC) .

IV. IMPLEMENTATION

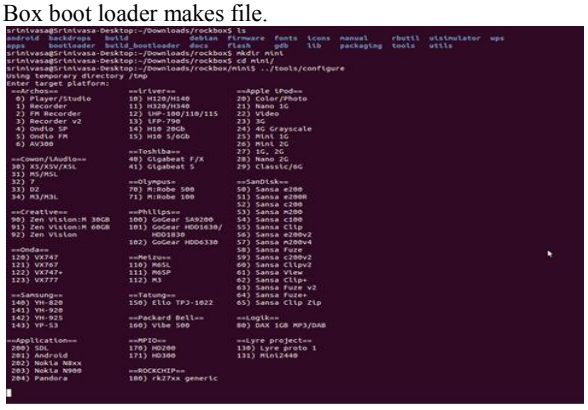
Steps for implementation:

1. nstall tool chain source code, download Virtual Box Ubuntu Image Disk and open the image with the use of Virtual Box. All required tool chain setup is from Rock Box source.
2. Download Source Code of Rock Box

$svncosvn://svn.rockbox.org/rockbox/trunk/home/user/rock box.

3. Compiling Rock Box for Mini2440:

- Create a Directory mini_boot
  – $mkdirmini_boot
  – $cd mini_boot
  – $../tools/configure
- Type 131 and select " B ".on the screen 131 shows mini2440 board
  "B" option creates Rock Box boot loader makes file.



4. Now type:

– $make

5. It will create a "bootloader.bin" file. This is useful for us when we run project on the selected board, which in turn helpful to boot the selected board. After getting the bootloader file we need to compile the GUI for Rock Box. In order to compile the GUI we need to follow some steps with the help of the following commands [9].

– $cd
– $mkdirmini_build
– $cd mini_build

– $../tools/configure
- Type 131 and select "N" .on the screen we can see 131 belongs to mini2440 board.
- "N" option creates make file for Normal mode.

We need to create a make file and we need to make it as a zip file to dump the zip file into mini2440 board in a extracted format, for this we need to use following commands.

– $make
– $make zip
– Now we need to copy "rockbox.zip" file in mini_build directory, because while booting it has to take some of the supporting files from this file only to create GUI for the board.
– We are ready with the binaries required to load on to the board. Now we need to connect the board "Mini2440" board to PC using HJATG cable provided with the board. This connection is required to dump the boot file in to the Mini2440 board. Connect one end of the cable to on-board JTAG port other end connect to PC 'COM' port.

LOADING BOOTLOADER:

Mini2440 by default contains "supervivi" bios in "NOR FLASH" .Now we are going to remove the default supervivi from the Nor Flash and we are loading the Rock Box bootloader file in to Nor FLASH. When we are loading bootloader file in to Nor Flash ensure that the board is in NOR mode (S2 should be in NOR side only) we need to install the HJTAG software which comes by default with the Mini2440 [9].

STEPS:

a) Open HJATG from program files (START→PROGRAMFILES) .
b) Switch on the board Mini2440.
c) Click on the HJATAG control and select 'check target'.
(Our board must be detected by the software)

Figure 4.10 MINI 2440 layout

d) Copy '.his' and 'hfc' files which are by default provided with the HJTAG software on DVD
e) Go to C: /PROGRAMFILES/HJTAG in the command prompt and after that go to 'init' (init script).
f) Click on 'Load' by selecting ".his" file in HJTAG installation directly, and then click on OK.
g) Go to H-Flasher, it will open a window
h) Click on "flash selection" (this option is present at the left side of the menu).
i) Select "SST39VF1601".
j) Click on "init scripts" and go to button '←' and click on 'cmd' column.
k) Select "soft reset" button and click up arrow button to first position.

(Note: This step is important because if we didn't select "soft reset" we could not run the program on "NOR FLASH").

Figure 4.11 MINI 2440 layout

l) Now click on "programming" on left side menu. Then click on 'click'.

----Here we need to see our NOR FLASH device i.e. AMD29LV160DB.

m) Select 'plain binary format' in "TYPE".
n) Select 'flash base address' in "Drt Address".
o) Select 'browse' and select 'rockboxbootloader' .i.e. "bootloader.bin" in Src File.

FLASH START ADDRESS ----------0X00000000
RAM START ADDRESS ----------0X40000000

p) Click on program
q) Now we can see a message "program successful".
r) Now connect the MICRO SD CARD to PC and format it by using "FAT32" file system.
s) Now copy the "rockbox.zip" file in to memory card and unzip the file. By unzipping it will get a folder with the name as ".rock box".
t) We need to copy the audio files which are in FLAC format in to the memory card along with the .rockbox folder and insert the memory card to the boards SD- CARD Slot.
u) Now 'RESET' the board.

After resetting bootloader file will be executed and it will create a GUI for the MINI2440 board and it must me appear then only the procedure will be successful. If we didn't insert the micro SD card in to the slot will get an error message as" PARTITION NOT FOUND" on the screen .

V. CONCLUSION

The Software implementation for Free Lossless Audio Decoding System with the support of Embedded Development Board using ARM-9 is done and verified with the help of FRIENDLY ARM BOARD (MINI 2440). There is more future scope for this project because everyone wants to hear quality music with a very attractive user interface and that is only possible only with the help of FLAC Decoder.

References

[1]. SanjuktaBhanja, N. Ranganathan, "Hardware Implementation of Data Compression,"*Lossless Compression Handbook, 2003, PP. 405-446*
[2]. AmitDhir, "Audio Players," *The Digital Consumer Technology Handbook, 2004, PP. 31-53*
[3]. Ida MengyiPu. "Audio compression," *Fundamental Data Compression, 2005, PP. 171-188*
[4]. J. Coalson, FLAC format, Xiph.Org Foundation Std. http://flac.sourceforge.net/format.html
[5]. David J. Katz, Rick Gentile, "Basics of Embedded Audio Processing," *Embedded Media Processing, 2006, PP.149-187*
[6]. S3C2440 Users' manual. http:// www.Samsung.com.
[7]. TDA1543 Data sheet. http://www.nxp.com.
[8]. CS8412 Data sheet. http://www.crystal.com.
[9]. http://flac.sourceforge.net/format.html.