



08-6316-RN-ZCH66
AUG 07, 2008
3.8

Release Notes for MP3 Decoder on ARM11 ELINUX

ABSTRACT:

Release Notes for MP3 Decoder on ARM11 ELINUX

KEYWORDS:

Multimedia codecs, MP3, MPEG

APPROVED:

Shang Shidong

Revision History

VERSION	DATE	AUTHOR	CHANGE DESCRIPTION
1.0	31-Aug-2004	Zakir Ahmed	Final release 1.0
1.1	29-Oct-2004	Zakir Ahmed	Release 1.1 with Debug Logging capability
2.0	09-Mar-2005	Zakir Ahmed	Hard Ware tested release for ARM11
2.1	01-Sep-2005	Anirudh Radhakrishnan	Build procedure changes for RVDS2.2
3.0	06-Feb-2006	Lauren Post	Using new format
3.1	09-Apr-2007	Srinidhi T N	Added support for Mpeg1 layer 2
3.2	04-May-2007	Madhav Varma	Updated information about Layer2 supporting sampling frequencies
3.3	22-May-2007	Surendra Jain P	Bug fixes
3.4	20-Jun-07	Jim Chacko	Added support for ARM9
3.5	29-Oct-2007	Terry Lv	Release of mp3d 1.08.
3.6	29-Jan-2008	Bing Song	Added support for MPEG-1/2 layer 1
3.7	03-Mar-2008	Bing Song	Added support for MPEG-2 layer 2
3.8	07-Aug-2008	Baofeng Tian	update for one code base and second round optimization

Table of Contents

Introduction	4
1.1 Purpose	4
1.2 Scope	4
1.3 Audience Description	4
1.4 References	4
1.4.1 Standards	4
1.4.2 General references	4
1.4.3 Freescale Multimedia References	5
1.5 Definitions, Acronyms, and Abbreviations	5
1.6 Document Location	6
2 Release History	7
2.1 Assumptions and Known Problems	8
2.2 Contacts	8
3 List of Deliverables	9
3.1 Documentation	9
3.2 Public Headers	9
3.3 Test Application Source	9
3.4 Library Source	9
3.5 Common Makefiles	10
3.6 Test Vectors	10
4 Software Setup & Tools used	12
5 Build Procedure	13
5.1 Library	13
5.2 Test Application	14
6 Test Application Execution	16
6.1 ELINUX	16
6.2 RVDS	16
6.3 UNIX Reference	16

Introduction

1.1 Purpose

The purpose of this document is to provide information on the package contents, instructions on building library and test applications and test execution on ARM11, RVDS and Linux x86.

1.2 Scope

The scope is restricted to information on the package contents and instructions for building and testing. This document does not provide architecture or details about the APIs provided in the package. Performance data will be provided in another document as detailed in the Requirements Book.

1.3 Audience Description

The reader is expected to have basic understanding of Audio Signal processing and MP3 decoding.

1.4 References

1.4.1 Standards

- ISO/IEC 11172-3:1993 Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s – Part 3: Audio (popularly known as *MPEG-1 Audio*).
- ISO/IEC 11172-4:1995 Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s – Part 4: Conformance testing (known as *MPEG-1 Conformance Testing*).
- ISO/IEC 13818-3:1998 Information technology – Coding of moving pictures and associated audio information – Part 3: Audio (popularly known as *MPEG-2 Audio LSF*).
- “MPEG Layer-3 Bitstream Syntax and Decoding” – by Ralph Sperschneider, issue – 1.3 dated 9th Sep 1997. (This document describes the bitstream syntax of of ISO/MPEG layer3 bitstream. This also includes syntax extension of MPEG Layer3 bitstream to meet the requirements of very low bitrates and sampling frequencies {8Khz, 11.025Khz, 12Khz}. These lower sampling frequencies syntax extension is not standardized by ISO is called MPEG-2.5)

1.4.2 General references

- Ted Painter and Andreas Spanias, “Perceptual Coding of Digital Audio”, Proc. IEEE, vol-88, no.4, april 2000
- H.S.Malvar, “Lapped transforms for efficient subband/transform coding”, IEEE trans. ASSP, June 1990.

- J.P.Princen, A.W.Johnson, A.B.Bradley, “Subband/transform coding using filterbank design based on time domain aliasing cancellation”, in proc. IEEE Int. conference ASSP, april1987
- MPEG Layer3 Bitstream syntax and decoding (Includes MPEG 2.5 layer3)
- “A Tutorial on MPEG/Audio compression” by Davis Pan

1.4.3 Freescale Multimedia References

- MP3 Decoder Application Programming Interface – mp3_dec_api.doc
- MP3 Decoder Requirements Book – mp3_dec_reqb.doc
- MP3 Decoder Test Plan – mp3_dec_test_plan.doc
- MP3 Decoder Release notes – mp3_dec_release_notes.doc
- MP3 Decoder Test Results – mp3_dec_test_results.doc
- MP3 Decoder Performance Results – mp3_dec_perf_results.doc
- MP3 Decoder Interface Header – mp3_dec_interface.h
- MP3 Decoder Application Code – mp3_dec_api.c

1.5 Definitions, Acronyms, and Abbreviations

TERM/ACRONYM	DEFINITION
AAC	Advanced Audio Coding
ADIF	Audio_Data_Interchange_Format
ADTS	Audio_Data_Transport_Stream
API	Application Programming Interface
ARM	Advanced RISC Machine
DAC	Digital to Analog Converter
FSL	Freescale
IEC	International Electro-technical Commission
ISO	International Standards Organization
LC	Low Complexity
MDCT	Modified Discrete Cosine Transform
MP3	Layer 3, MPEG2 Layer3 and MPEG2.5 Layer 3
MPEG	Moving Pictures Expert Group
OS	Operating System
PCM	Pulse Code Modulation
PNS	Perceptual Noise Substitution
RVDS	ARM RealView Development Suite
TBD	To Be Determined
UNIX	Linux PC x/86 C-reference binaries

1.6 Document Location

docs/mp3_dec

2 Release History

RELEASE NUMBER	DELIVERABLES	FEATURES
1.0		Engineering Release
2.0	<ul style="list-style-type: none"> • Documentation • Application Interface header file • ELINUX and RVDS libraries and test applications • UNIX/Linux x/86 Reference library and test application • Makefiles and Source code for library and test application including optimized assembler for the ELINUX and RVDS libraries. • Test vectors 	<ul style="list-style-type: none"> • Supports all sampling rates specified by ISO, and MPEG2.5 extensions. (MPEG1:32/44.1/48Khz, MPEG2-LSF:16/22.05/24Khz, MPEG2.5: 8/11.025/12Khz) • Supports all bitrates specified by ISO (MPEG1 : 32-320kbps, MPEG-2 LSF : 8-160kbps) • Stereo / Mono decoding
2.1	Same	<ul style="list-style-type: none"> • Shared Library Support • Upgrade to RVDS 2.2
2.2	Same	<ul style="list-style-type: none"> • Fix for UNIX bitmatch
2.3	Same	<ul style="list-style-type: none"> • Added feature for Mpeg 1 layer 2. • Error protection (CRC is not implemented) , Supports sampling frequencies of 32kHz, 44.1kHz and 48kHz only. i.e.,MPEG2-LSF not supported. • Tested for RVDS, Elinux and UNIX
2.4	Same	<ul style="list-style-type: none"> • Fixed the rewindNbits call in the Huffman_dequant () (Test streams: s_helix.mp3, 1_daolang.mp3, MU.mp3) • Fixed Alignment of Scalefactor bands, from Big_ values to Count1 region transition (Test streams: friends_48khz_128kbps.mp3, c03.mp3, 01_Me_Black_Me.mp3, a04_44khz_cbr128.mp3, a05_44khz_cbr128.mp3, a06_44khz_vbr128.mp3, e01_44khz_cbr_128.mp3) • Fixed the resetting of quadruples in Huffman_dequant() resulting in segmentation fault (Test stream: Angel_Eyex.mp3)

2.5	Same	<ul style="list-style-type: none">• Integrate mp2 QMFS optimizations.• Integrate FastDCT and IMDCT_LEE c code.
2.6	Same	<ul style="list-style-type: none">• Added feature for MPEG-1/2 layer 1
2.7	Same	<ul style="list-style-type: none">• Added feature for MPEG-2 layer 2

Table 1. Details of the release

2.1 Assumptions and Known Problems

None

2.2 Contacts

Please report any problems to Freescale customer representative.

3 List of Deliverables

3.1 Documentation

Base directory: /vobs/fsl_mad_multimedia_codecs/

Subdirectory	Files
docs/mp3_dec	mp3_dec_api.doc mp3_dec_reqb.doc mp3_dec_test_plan.doc mp3_dec_test_results.doc mp3_dec_perf_results.doc mp3_dec_release_notes.doc

3.2 Public Headers

Base directory: /vobs/fsl_mad_multimedia_codecs/

Subdirectory	File
ghdr	mp3_dec_interface.h

3.3 Test Application Source

Base directory: /vobs/fsl_mad_multimedia_codecs/test/

Subdirectory	Files
mp3_dec /	“Makefile” makefile for building RVDS, UNIX and ELINUX board executables.
mp3_dec /c_src	*.c, application code.
mp3_dec /hdr	*.h, application header files

3.4 Library Source

Base directory: /vobs/fsl_mad_multimedia_codecs/src/

Subdirectory	Files
mp3_dec	Makefile “Makefile” for building RVDS, UNIX, and ELINUX libraries. libmp3_dec_arm11_RVDS.a – Special options for simulator testing libmp3_dec_arm11_ELINUX.a - static library for board libmp3_dec_arm11_ELINUX.so – shared library for board libmp3_dec_UNIX.a – library for Linux x/86 – c reference code
mp3_dec /c_src	*.c, MP3 decoder source code
mp3_dec /asm_arm	*.s MP3 assembler source
mp3_dec /hdr	*.h, MP3 decoder library header files

3.5 Common Makefiles

Base Directory: /vobs/fsl_mad_multimedia_codecs/build

Subdirectory	Files
Makefile.init	<p>This is a common makefile included in the codec library makefile for building the libraries. This file includes common options used by all codecs. Following flags can be overwritten or added to in the codec library makefile</p> <ol style="list-style-type: none"> 1. Path to toolchain tools (TC_ROOT) 2. GNU header file path (HEADER_PATHS) 3. GNU library path (LIB_PATHS) 4. GNU Compiler/Assembler Options (GNU_CFLAGS, GNU_AFLAGS) 5. Endian Flags 6. Optimization Flags(OPTIM_LEVEL, OPTIM_TYPE) 7. Common options for RVDS,UNIX and ELINUX (CFLAGS,AFLAGS) 8. Build specific flags 9. Source directory of 'C' code 10. Source directory of 'assembly(.s)' code 11. Object directory for .o files 12. RVDS Compilation Tools 13. Codec header path 14. Arguments for librarian for UNIX builds 15. SHARED_ELINUX builds for libraries that must be linked using the toolchain because of external library includes.
Makefile_test.init	<p>This is the common makefile included in the codec test makefile for building the test application. This file includes the common options used by the all the codecs. Following flags can be overwritten or added to in the codec test makefile</p> <ol style="list-style-type: none"> 1. Toolchain path depending on the build option 2. Compiler Flags 3. Linker flags 4. Paths for c_source, exe and object directories 5. Codec header files' INCLUDES path 6. Endian Flags 7. CODEC_LIB generation

3.6 Test Vectors

Base Directory: /vobs/multimedia_vectors/test_vectors

The test vectors are provided in another location from the library and test source.

Subdirectory	Files
mp3_dec/input	Input vectors.
mp3_dec/ref	reference vectors (floating point based)

4 Software Setup & Tools used

- ARM RVDS 2.2 (build 503) should be installed in the PC.
- Freescale Linux OS Release L26.1.15 must be running on the evaluation board.
- Intel based Red Hat Linux Machine must have the Montavista toolchain installed on it.
 - MontaVista 3.4.3-25.0.36.0501313 2005-08-21
- ‘Cygwin’ **Version** CYGWIN_NT-5.1, a freely downloadable linux emulator is installed in PC - <http://www.cygwin.com/>.
- ‘make’ utility available for targeted platforms

5 Build Procedure

All the required makefiles are provided under individual directories. The library can be built for windows / target processor (ARM1136J-S). The details for the build procedure are described below.

5.1 Library

To build the library, run 'make' on 'Makefile' from library directory. The makefile shall create the required directory to hold the object files. The makefile can be used if you want to build the library only. The same makefile can be used to build libraries for both board, Unix/Linux and RVDS with different build options. The following options are available to build the library.

Options

a) BUILD options:

- a. **BUILD=ARM11ELINUX** : This is the default option and builds both static library 'lib_mp3_dec_arm11_elinux.a' (for ARM11) and shared library 'lib_mp3_dec_arm11_elinux.so' (for ARM11) for testing on the board.
- b. **BUILD=ARM11LERVDS**: This option builds the static library 'lib_mp3_dec_arm11_lervds.a' (for ARM11) for testing on RVDS (Armulator).
- c. **BUILD=UNIX**: This option builds the static library 'libmp3_dec_UNIX.a', for testing on UNIX/Linux machine.

Eg: make BUILD= ARM11ELINUX
 make BUILD=ARM11LERVDS
 make BUILD=UNIX

b) clean options:

- o **clean_make=ARM11LERVDS**: Deletes all the object files and the RVDS library 'lib_mp3_dec_arm11_lervds.a' (for ARM11).
- o **clean_make=ARM11ELINUX**: Deletes all object files and the ELINUX libraries lib_mp3_dec_arm11_ELINUX.a (for ARM11) and lib_mp3_dec_arm11_elinux.so (for ARM11).
- o **clean_make=UNIX**: Deletes all the object files and the UNIX library 'lib_mp3_dec_unix.a'.
- o **clean**: Deletes all the object files and RVDS, UNIX and ELINUX libraries.

Note: Make appropriate changes in file 'makefile.init' at directory 'ARM11/common' for the location of toolchains.

The library that is built is saved as lib_mp3_dec_arm11_lervds.a for ARM 11 RVDS build, and lib_mp3_dec_arm11_elinux.a and lib_mp3_dec_arm11_elinux.so for ARM11 board build. These libraries are saved in the current directory (the same directory in which the source and assembly directories are listed).

For ARM11

Target	Compilation Environment	Build Options	Library Name
Board	PC (Using Cygwin)	BUILD=ARM11ELINUX	Lib_mp3_dec_arm11_elinux.a lib_mp3_dec_arm11_elinux.so
RVDS	PC (Using Cygwin)	BUILD=ARM11LERVDS	Lib_mp3_dec_arm11_lervds.a
Unix/Linux	Linux/Unix machine	BUILD=UNIX	libmp3_dec_unix.a

5.2 Test Application

To build the test application, run 'make' on 'Makefile' from the test directory. This makefile can create executables for testing on both board and RVDS for ARM11. The executables, for ARM11, test_mp3_dec_arm11_lervds for RVDS, test_mp3_dec_arm11_elinux for board and test_mp3_dec_unix for UNIX are stored under test/exe directory. The makefile shall create the required directory structure to hold the object files and executables. The following commands should be invoked so as to build the executables.

Options

1) BUILD options:

- **BUILD=ARM11ELINUX:** This is the default option and builds the executable 'test_mp3_dec_arm11_elinux', for the board ARM11.
- **BUILD=ARM11LERVDS:** This option builds the executable 'test_mp3_dec_arm11_lervds' for the RVDS (Armulator) ARM11.
- **BUILD=UNIX:** This option builds the executable 'test_mp3_dec_unix' for the Unix/Linux machine.

Eg: make BUILD=ARM11ELINUX (for board)
 make BUILD=ARM11LERVDS (for Armulator)
 make BUILD=UNIX (for Unix/Linux machine)

2) LIBRARY options:

- **LIB=STATIC:** This option builds the ELINUX test application linked with the ELINUX static library 'lib_mp3_dec_arm11_elinux.a' for ARM11. If nothing is specified, the executable links with shared library 'lib_mp3_dec_arm11_elinux.so'
- Eg:** make LIB=STATIC

3) PROFILE options:

TIME_PROFILE=1 is used to get cycle measurement information.

4) clean options:

- **clean_BUILD=ARM11LERVDS:** Deletes all the object files and the RVDS ARM11 executable 'mp3_dec_arm11_RVDS' for ARM11.

- .
- **clean_BUILD=ARM11ELINUX:** Deletes all the object file and the ELINUX 'mp3_dec_arm11_ELINUX' for ARM11.
- **clean_UNIX:**Deletes all the object files and the Unix/Linux executable 'mp3_dec_UNIX'.
- **clean:**Deletes all the object files and RVDS,UNIX ELINUX executables.

Note:

In 'makefile_test.init' at directory 'ARM11/common', the paths for the compiling and linking tools are hard coded for the current set-up. These paths may not be the same in the user's directory set up. Hence, the 'makefile_test.init' should be modified to point to the directories where the linking and compilation tools are present before building the application for board.

The following table summarises the build options,

For ARM11

Target	Compilation Environment	Build Options	Executable Name
Board	Redhat Linux Machine	BUILD=ARM11ELINUX LIB= STATIC	test_mp3_dec_arm11_elinux
RVDS	PC (Using Cygwin)	BUILD=ARM11LERVDS	test_mp3_dec_arm11_lervds
UNIX/Linux	Unix/Linux machine	BUILD=UNIX	test_mp3_dec_unix

6 Test Application Execution

To run the scripts defines 'SINGLE_VECTOR' and 'ARM_ADS' has to be removed while building the library and the executable. This can be done by commenting the same at the makefiles. The shared library has to be in the current directory while running the executable built with shared library. Comparison of the outputs with the reference can be done using 'dif32_linux'. 'dif32_linux' can be run only on linux environment.

6.1 ELINUX

For ARM 11: *test_mp3_dec_arm11_elinux* <input vector> <output vector>

The output vector will be placed into files <output_vector >

6.2 RVDS

Please refer ARM documentation regarding loading the image and configuring the RVDS debugger for ARM1136J-S for ARM11.

- RVDS :
Once the image is loaded press "F5" or select the pull down menu option "*Debug -> Execution Control*" to run the loaded image.

6.3 UNIX Reference

To execute on Linux x/86 type:

***mp3_dec_UNIX* <input vector> <output vector>**

