08-6336-RN-ZCH66

8/31/2007

3.0

Release Notes for WMA10 Decoder on ARM11 ELINUX

ABSTRACT:

Release Notes for WMA10 Decoder on ARM11 ELINUX

KEYWORDS:

Multimedia codecs, WMA, Windows Media Audio

APPROVED:

Shang Shidong

Revision History

VERSION	DATE	AUTHOR	CHANGE DESCRIPTION
1.0-D01	19-July-2007	Sukruth K	Initial Draft
1.0	25-July-2007	Sukruth K	Initial version after review comments updatation
2.0	31-Aug-2007	Huaping Liu	Update for optimization version
3.0	15-Oct-2008	Lyon Wang	Update for Macro Define change

Table of Contents

1 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 ELINUX, 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 WMA decoding.

1.4 References

1.4.1 Standards

- "An overview of Window Media Audio Decoding", Microsoft Corporation
- ASF Specification from Microsoft Corporation, Revision 01.20.02, June 2004
- WMA Audio Concepts Microsoft Corporation
- WMA decoder block diagram", Microsoft Corporation

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.

1.4.3 Freescale Multimedia References

- WMA10 Decoder Requirements Book – wma10_dec_reqb.doc
- WMA10 Decoder Release notes – wma10_dec_release_notes.doc
- WMA Decoder Interface header – wma10_decoder_interface.h
- WMA Decoder Application Code – wma10_app_test.c

1.5 Definitions, Acronyms, and Abbreviations

TERM/ACRONYM	DEFINITION
API	Application Programming Interface
ARM	Advanced RISC Machine
ASF	Advanced Streaming Format
FSL	Freescale
OS	Operating System
PCM	Pulse Code Modulation
RVDS	ARM RealView Developer Suite
WMA	Windows Media Audio
UNIX	Linux PC x/86 C-reference binaries
TBD	To Be Done

1.6 Document Location

docs/wma10_dec

2 Release History

RELEASE NUMBER	DELIVERABLES	FEATURES
1.0	• 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	• Combo package consisting of WMA10 Standard, Professional & Lossless. • Tested the WMA10 package only with WMA Pro test vectors
2.0	• WMA Pro & Losslss optimization C and ASM code • Update some WMA Pro & Losslss test vectors Performance testing report	• Tested the WMA10 package with WMA Pro all test vectors, Lossless test vectors •

Table 1. Details of the Release

2.1 Assumptions and Known Problems

2.2 Assumptions

Combo pack containing WMA10 Standard, Professional & Lossless Decoder has been tested for WMA Pro & Lossless test vectors .

2.3 Known Problems

- Re-entrancy & Re-location tests have not been performed
- Not tested for Big Endian support
- Microsoft certification for WMA Professional shall be done

2.4 Contacts

Please report any problems to Freescale customer representative

3 List of Deliverables

3.1 Documentation

Base directory: /vobs/multimedia_codecs/

Subdirectory	**Files**
docs/wma10_dec	wma10_dec_api.doc wma10_dec_reqb.doc wma10_dec _release_notes.doc

3.2 Public Headers

Base directory: /vobs/multimedia_codecs/

Subdirectory	**File**
ghdr/wma10_dec	*.h, application interface header files

3.3 Test Application Source

Base directory: /vobs/multimedia_codecs/test/

Subdirectory	**Files**
wma10_dec/	"Makefile" makefile for building RVDS, UNIX and ELINUX board executables.
wma10_dec/c_src	*.c, application code.
wma10_dec /hdr	*.h, application header files

3.4 Library Source

Base directory: /vobs/multimedia_codecs/src

Subdirectory	**Files**
wma10_dec	Makefile "Makefile" for building RVDS, UNIX, and ELINUX libraries. lib_wma10_dec_arm11_LERVDS.a – For Little Endian RVDS lib_wma10_dec_arm11_ELINUX.a - static library for board lib_wma10_dec_arm11_ELINUX.so – shared library for board lib_wma10_dec_x86_UNIX.a – library for Linux x/86 – c reference code
wma10_dec /c_src	*.c, WMA10 decoder source code
wma10_dec /asm_arm	*.s WMA10 decoder assembly source
wma10_dec /hdr	*.h, WMA10 decoder library header files

3.5 Common Makefiles

Base Directory: /vobs/multimedia_codecs/build

Makefile	**Description**
Makefile.init	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 1. Path to toolchain tools (TOOLS_DIR) 2. Header file path 3. Library path (SYS_INCLUDE) 4. Endian Flags 5. Optimization Flags(OPTIM_LEVEL, OPTIM_TYPE) 6. Common options for RVDS,UNIX and ELINUX (CFLAGS,AFLAGS) 7. Build specific flags 8. Source directory of 'C' code 9. Source directory of 'assembly(.s)' code 10. Object directory for .o files 11. RVDS Compilation Tools 12. Codec header path 13. Arguments for librarian for UNIX builds 14. SHARED_ELINUX builds for libraries that must be linked using the toolchain because of external library includes.
Makefile_test.init	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 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

3.6 Test Vectors

/vobs/multimedia_vectors/wma10_dec

1.input

2.ref

4 Software Setup & Tools used

- ARM RVDS 2.2 (build 503) should be installed in the PC.
- Freescale Linux OS Release L26.1.16 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
- ‘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 the 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 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_wma10_dec_arm11_ELINUX.a' and shared library 'lib_wma10_dec_arm11_ELINUX.so' , for testing on the board.

b. **BUILD=ARM11LERVDS**: This option builds the static library 'lib_wma10_dec_arm11_LERVDS.a', for testing on RVDS (Armulator) in Little Endian mode.

Eg:

```
make BUILD=ARM11ELINUX
make BUILD=ARM11LERVDS
```

b) **clean options**:

○ **clean**: Deletes all the object files and libraries. To be used with the BUILD option.

Note: Make appropriate changes in file 'makefile.init' at directory '/vobs/multimedia_codecs/build' for the location of toolchains.

The library that is built is saved as lib_wma10_dec_arm11_LERVDS.a for LERVDS build, and lib_wma10_dec_arm11_ELINUX.a and lib_wma10_dec_arm11_ELINUX.so for board build. These libraries are saved in the current directory (the same directory in which the source and assembly directories are listed).

Target	Compilation Environment	Build Options	Library Name
Board	Redhat Linux Machine	BUILD= ARM11ELINUX	lib_wma10_dec_arm11_ELINUX.a lib_wma10_dec_arm11_ELINUX.so
RVDS	Redhat Linux Machine	BUILD=LERVDS	lib_wma10_dec_arm11_LERVDS.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 test_wma10_dec_arm11_LERVDS for RVDS, test_wma10_dec_arm11_ELINUX for board are stored under test/wma10_dec 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_wma10_dec_arm11_ELINUX’, for the board.

   - **BUILD=ARM11LERVDS**: This option builds the executable ‘test_wma10_dec_arm11_RVDS’ for the RVDS (Armulator).

   Eg:

   ```
   make BUILD=ARM11ELINUX  (for board)
   make BUILD=ARM11LERVDS     (for Armulator)
   ```

2) **LIBRARY options**:
   - **LIB_TYPE= STATIC**: This option builds the ELINUX test application linked with the ELINUX static library ‘lib_wma10_dec_arm11_ELINUX.a’.If nothing is specified ,the executable links with shared library ‘lib_wma10_dec_arm11_ELINUX.so’

     Eg: make LIB=STATIC

3) **clean options**:
   - **clean**: Deletes all the object files and executables. To be used along with build options.

<u>Note:</u>

In ‘makefile_test.init’ at directory ‘/vobs/multimedia_codecs/build’, 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,

Target	Compilation Environment	Build Options	Executable Name

Board	Redhat Linux Machine	BUILD=ARM11ELINUX
		LIB_TYPE= STATIC
			test_wma10_dec_arm11_ELINUX

RVDS	Redhat Linux Machine	BUILD=ARM11LERVDS	test_wma10_dec_arm11_RVDS

6 Test Application Execution

6.1 Scripts

TBD

6.2 ELINUX Single

Test_wma10_dec_arm11_ELINUX <input vector> <output vector>

The output vector will be placed into files <output_vector>.

6.3 RVDS

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

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

6.4 Conformance criteria

Verification of outputs:

The following executables used to verify the outputs.

- ansiwmadecs10full.exe – Standard decoder from Microsoft to generate the reference vectors.

  Usage: ansiwmadecs10full.exe <input_vector> <output_vector_ref>

- wavdiff24.exe – 24 Bit Comparator

  Usage: wavdiff24.exe <output_vector> <output_vector_ref>

Please refer test plan document for Microsoft conformance criteria.

Note: Microsoft certification for Professional shall be done.

7 Pre compilation Options

7.1 Test Application compilation Options

The following C options need to be set

Application	Description	Remarks
• DISCARD_OUTPUT	To disable dumping of output.	All builds except UNIX
• TEST_PERFORMANCE	To disable dumping of output while taking performance	All builds except UNIX
• TIME_PROFILE	To enable performance test	Only for ELINUX
• TIME_PROFILE_RVDS	To enable performance test	Only for RVDS

7.2 Library compilation options

Library	Desciption	Remarks
• BUILD_INTEGER • ENABLE_LPC • ENABLE_ALL_ENCOPT • DISABLE_OPT • BITSTREAM_CORRUPTION_ASSERTS • WMA_DECPK_BUILD • WMAAPI_NO_DRM • WMAAPI_NO_DRM_STREAM • DISABLE_UES • _CRT_NON_CONFORMING_SWPRINTFS • _CRT_SECURE_NO_DEPRECATE	Always Enabled	All Builds
• OPT_ASM=1 • WMAPD_QL2=1 -DWMAPD_QL1=01	Enabled to use optimized C and ASM code	ARM Platform

- OPT_ASM=0	Disabled for ARM Platform	ARM Platform