08-7224-RN-ZCH66

10/27/2008

4.0

Release Notes for WB-AMR Decoder and Encoder

ABSTRACT:

Release Notes for WB-AMR Decoder and Encoder

KEYWORDS:

Multimedia codecs, WBAMR, speech

APPROVED:

Shang shidong

Revision History

VERSION	DATE	AUTHOR	CHANGE DESCRIPTION
1.0	25-Jan-2005	Ashok Kumar	Final release of WB-AMR codec on RVDS
2.0	28-Apr-2005	Ashok Kumar	Final release of WB-AMR codec on RVDS
2.1	12-Sep-2005	Anirudh Rahakrishnan	Build procedure changes for RVDS2.2
3.0	06-Feb-2006	Lauren Post	Using new format
3.1	15-Mar-2006	Gaurav Goel	Addition of Review changes
3.2	19-Jan-2007	Sukruth	Update for ARM9 RVDS Big Endian Support
4.0	27-Oct-2008	Jackiea Pan	Update document ID

Table of Contents

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 ELINUX, RVDS and WinCE.

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 Speech Signal processing and WB-AMR vocoder.

1.4 References

1.4.1 Standards

- **3GPP TS 26.190 V5.1.0 (2001-12)**: AMR Wideband Speech Codec; Transcoding functions (Release 5)
- **3GPP TS 26.191 V5.1.0 (2002-03)**: AMR Wideband Speech Codec; Error Concealment of Lost Frames (Release 5)
- **3GPP TS 26.192 V5.0.0 (2001-03)**: AMR Wideband Speech Codec; Comfort Noise Aspects (Release 5)
- **3GPP TS 26.193 V5.0.0 (2001-03)**: AMR Wideband Speech Codec; Source Controlled Rate operation (Release 5)
- **3GPP TS 26.194 V5.0.0 (2001-03)**: AMR Wideband Speech Codec; Voice Activity Detection (VAD) (Release 5)
- **3GPP TS 26.174 V5.4.0 (2002-12)**: AMR Wideband Speech Codec; Test sequences (Release 5)
- **3GPP TS 26.173 V5.8.0 (2003-09)**: AMR Wideband Speech Codec; ANSI-C code (Release 5)
- **3GPP TS 26.201 V5.0.0** (2001-03): AMR Wideband Speech Codec; Frame Structure.
- **ITU-T Recommendation G.711 (1988)** – Coding of analogue signals by pulse code modulation Pulse code modulation (PCM) of voice frequencies.

1.4.2 General references

- E. Paksoy, J.C.D Martin, Alan McCree, C.G.Gerlach, Anand Anandakumar, Wai-Ming Lai and Vishu Viswanathan, ICASS Proceeding 1999 “An Adaptive Multi-Rate Speech Coder for Digital Cellular Telephony”
- Real-Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs “RFC3267”

1.4.3 Freescale Multimedia References

- WB AMR Codec Application Programming Interface – wbamr_codec_api.doc
- WB AMR Codec Requirements Book – wbamr_codec_reqb.doc
- WB AMR Codec Test Plan - wbamr_codec_test_plan.doc
- WB AMR Codec Release notes - wbamr_codec_release_notes.doc
- WB AMR Codec Test Results – wbamr_codec_test_results.doc
- WB AMR Codec Performance Results – MX31_performance.xls
- WB AMR Codec Performance Results – MX21_performance.xls
- WB AMR Codec Performance Results – RVDS_ARM11_LE_performance.xls
- WB AMR Codec Performance Results – RVDS_ARM9_LE_performance.xls
- WB ARM Interface Common Header – wbamr_common_interface.h
- WB ARM Interface Decoder Header – wbamr_dec_interface.h
- WB ARM Interface Encoder Header – wbamr_enc_interface.h
- WB ARM Decoder Application Code – wbamr_dectest.c
- WB ARM Encoder Application Code – wbamr_enctest.c

1.5 Definitions, Acronyms, and Abbreviations

TERM/ACRONYM	DEFINITION
3GPP	3RD GENERATION PARTNERSHIP PROJECT
ACELP	ALGEBRAIC CODE EXCITED LINEAR PREDICTION
API	APPLICATION PROGRAMMING INTERFACE
ARM	ADVANCED RISC MACHINE
CNG	COMFORT NOISE GENERATION
DTX	DISCONTINUOUS TRANSMISSION
ETSI	EUROPEAN STANDARD TELECOMMUNICATIONS SERIES
FSL	FREESCALE
IF-1	INTERFACE FORMAT 1
IF-2	INTERFACE FORMAT 2

ITU	INTERNATIONAL TELECOMMUNICATION UNION
LSP	LINE SPECTRAL PAIR
LP	LINEAR PREDICTION
MIME	MULTIMEDIA STORAGE FORMAT
MIPS	MILLION INSTRUCTIONS PER SECOND
OS	OPERATING SYSTEM
PCM	PULSE CODE MODULATION
RVDS	ARM REALVIEW DEVELOPMENT SUITE
SCR	SOURCE CONTROLLED RATE
SID	SILENCE INSERTION DESCRIPTOR
TBD	TO BE DETERMINED
UNIX	LINUX PC X/86 C-REFERENCE BINARIES
WB-AMR	WIDE BAND ADAPTIVE MULTI-RATE CODEC
VAD	VOICE ACTIVITY DETECTION

1.6 Document Location

docs/wb_amr

2 Release History

RELEASE NUMBER	DELIVERABLES	FEATURES
1.0		• Engineering Release
2.0	• Documentation • Interface header file for encoder and decoder • C source and ASM files • Library Makefile	• Contains prototypes of interface function and data types • Details of feature and interface function can be found in these docs • Optimized C and asm files • makefile can be used to generate libraries
2.1	• test vectors	• Contains standard (3GPP) and non standard (generated using C reference code) input and reference vectors for encoder and decoder. IF-1, IF-2 and MMS format test vectors are also provided
2.2	• Sample test application (for encoder and decoder), makefile and script to run test in batch mode	• All the feature of previous release is present in this release also. Some more scripts are added to test codec on board/ELINUX
2.3	Same plus • 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.	• Shared Library Support • Upgrade to RVDS 2.2 • Bus Alignment Fixes • Name Collision Fixes
2.5	Same	• CR fix for Wrong SID frame size in IF1
2.6	Same	• ARM9 Optimizations and support for Big Endian on ARM9.

Table 1. Details of the release

2.1 Assumptions and Known Problems

- Application needs to call separate function (defined in library) to relocate encoder or decoder table to different memory location. Please refer encoder/decoder test application to see the details of this testing.

2.2 Contacts

Please report any problems to Freescale customer representative.

3 List of Deliverables

3.1 Documentation

Base directory: / fsl_mad_multimedia_codecs/

Directory	Files
docs/wb_amr	wbamr_codec_api.doc wbamr_codec_reqb.doc wbamr_codec_test_plan.doc wbamr_codec_test_results.doc wbamr_codec _release_notes.doc wbamr_codec_perf_results.doc wbamr_codec _datasheet.doc

3.2 Public Headers

Base directory: / fsl_mad_multimedia_codecs /

Directory	Files
ghdr	wbamr_common_api.h wbamr_enc_api.h wbamr_dec_api.h

3.3 Test Application Source

Base directory: / fsl_mad_multimedia_codecs/

Subdirectory	Files
test/wb_amr	"Makefile" makefile for building RVDS, UNIX and ELINUX board executables.
test/wb_amr/hdr	*.h, application headers.
test/wb_amr/c_src	*.c, application code.
utils/wb_amr	Batch files to be run on the board and RVDS

3.4 Library Source

Base directory: / fsl_mad_multimedia_codecs/

Subdirectory	Files
src/wb_amr	Makefile "Makefile" for building RVDS, UNIX, and ELINUX libraries. lib_wb_amr_dec_arm9_elinux.a: decoder static library for ARM926EJ-S lib_wb_amr_dec_arm9_elinux.so: decoder dynamic library for ARM926EJ-S lib_wb_amr_dec_arm9_lervds.a: decoder ARM9 LERVDS library

	lib_wb_amr_enc_arm9_elinux.a: encoder static library for ARM926EJ-S lib_wb_amr_ enc _arm9_elinux.so: encoder dynamic library for ARM926EJ-S lib_wb_amr_ enc _arm9_lervds.a: encoder ARM9 LERVDS library
src/wb_amr/c_src	*.c, WB_AMR source code
src/wb_amr/hdr	*.h WB_AMR library header files

3.5 Common Makefiles

Base Directory: / fsl_mad_multimedia_codecs/

Makefile	Description
build/Makefile.init	This is a common makefile. To build libraries, it is included in the codec library makefile. This file includes common options used by all codecs.
build/ Makefile_test.init	This is the common makefile included in the codec test makefile building the test application. This file includes the common opti used by the all the codecs.

3.6 Test Vectors

Base Directory: multimedia_vectors/test_vectors

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

Directory	Files	Description
wb_amr/test_vectors	Input and reference test vectors for encoder and decoder are given in this folder	IF-1, IF-2 and MMS reference vectors generated using pure C code are also given in this folder

4 Software Setup & Tools used

- ARM RVDS 3.02 (build 586) 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 (ARM926EJ-S). The details for the build procedure are described below.

Note: The build procedure is explained with decoder as an example.To build library for the encoder applies the same procedure given below, with the makefile.

5.1 Library

To build the library, run 'make' on 'Makefile' from src/wb_amr directory. This makefile can create libraries for testing on ARM board, RVDS, Linux and UNIX. 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 following options can be invoked so as to build the library

Options

a) **BUILD options**:

- **BUILD= ARM11ELINUX** : It builds both static as well as dynamic libraries, 'lib_wb_amr_dec_arm11_elinux.a ' and shared library 'lib_ wb_amr_dec_arm11_elinux.so' , for testing on the board.

- **BUILD=ARM11LERVDS**: This option builds the static library 'lib_wb_amr_arm11_lervds.a', for testing on ARM11 LE RVDS (Armulator).

- **BUILD= ARM9ELINUX:** It builds static library, 'lib_wb_amr_arm9_elinux.a' for testing on the board.

- **BUILD=ARM9LERVDS**: This option builds the static library 'lib_wb_amr_arm9_lervds.a', for testing on ARM9 LE RVDS (Armulator).

b) **clean options**:

- **clean**: Deletes all the object files and the library for specified BUILD option.

Note: Make appropriate changes in file 'Makefile.init' for the location of toolchains.

The libraries are saved in the current directory, src/wb_amr.

Target	Compilation Environment	Build Options	Library Name
Board (MX31)	PC(Using Cygwin)	BUILD=ARM11ELINUX	lib_wb_amr_dec_arm11_elinux.a lib_wb_amr_enc_arm11_elinux.a lib_wb_amr_dec_arm11_elinux so lib_wb_amr_enc_arm11_elinux.so
RVDS	PC(Using Cygwin)	BUILD=ARM11LERVDS BUILD=ARM9LERVDS	lib_wb_amr_dec_arm11_lervds.a lib_wb_amr_enc_arm11_lervds.a lib_wb_amr_dec_arm9_lervds.a

			lib_wb_amr_enc_arm9_lervds.a
Unix/ Linux	Unix/Linux machine	BUILD=UNIX	lib_wb_amr_x86_dec_unix.a lib_wb_amr_x86_dec_unix.a
Board (MX21)	Linux/Unix machine	BUILD= ARM9ELINUX	lib_wb_amr_dec_arm9_elinux.a, lib_wb_amr_enc_arm9_elinux.a

5.2 Test Application

To build the test application, run 'make' from the test/wb_amr directory. This makefile can create executables for testing on Linux x86, the ARM11/ARM9 board and RVDS for ARM11.. The following commands should be invoked so as to build the executables.

Note: The build procedure is explained with decoder as an example.To build library for the encoder applies the same procedure given below, with the makefile 'Makefile'.

Options

1) **BUILD options**:
   - **BUILD=ARM11ELINUX**: This option builds the executable 'test_wb_amr_arm11_elinux', for MX31 board.
   - **BUILD=ARM11LERVDS**: This option builds the executable 'test_wb_amr_arm11_lervds ' for the ARM11 LE RVDS (Armulator).
   - **BUILD=ARM9ELINUX**: This option builds the executable 'test_wb_amr_arm9_elinux', for MX21 board.
   - **BUILD=ARM11LERVDS**: This option builds the executable 'test_wb_amr_arm9_lervds ' for the ARM11 LE RVDS (Armulator).
   - **BUILD=UNIX**: This option builds the executable 'test_wb_amr_x86_unix' for the Unix/Linux machine.

2) **LIBRARY options**:
   - **LIB_TYPE= STATIC**: This option builds the ELINUX test application linked with the ELINUX static library 'lib_wb_amr_arm11_elinux.a'.If nothing is specified ,the executable links with shared library 'lib_wb_amr_arm11_elinux.so'

     Eg: make BUILD=ARM11ELINUX LIB_TYPE=STATIC

3) **clean options**:
   - **clean:** Deletes all the object files and executable for the specified BUILD option

Note:

In 'Makefile_test.init', 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, it 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 (MX31)	Redhat Linux Machine	BUILD=ARM11ELINUX LIB_TYPE = STATIC	test_wb_amr_dec_arm11_elinux test_wb_amr_enc_arm11_ elinux
RVDS	PC (Using Cygwin)	BUILD=LERVDS	test_wb_amr_dec_arm11_lervds test_ wb_amr _enc_arm11_ lervds test_wb_amr_dec_arm9_lervds test_ wb_amr _enc_arm9_ lervds
Board (MX21)	Redhat Linux Machine	BUILD=ARM9ELINUX	test_wb_amr_dec_arm9_elinux test_wb_amr_enc_arm9_elinux

6 Test Application Execution

6.1 Scripts

In the utils/wb_amr directory, a script file exists for doing

a) Regression, Performance on i.MX31 and i.MX27 (wb_amr_run_linux.sh)

b) Sanity on LE RVDS (wb_amr_run_rvds.sh)

The user is expected to be aware of the settings to be done for the hardware and to get Linux running on ARM11/ARM9

a) Go to the directory utils/wb_amr" and edit scripts verify that paths are correct.
b) Make sure the scripts are changed according to current test setup.
c) create a working directory on the board and copy the executables from test/wb_amr to the current directory
d) copy the required script file (.sh) from utils/wb_amr into the working directory on the board
e) Compare output of encoder and decoder using diff script provided in utils/wb_amr.

6.2 RVDS

The batch files to test encoder and decoder on RVDS are provided in utils/wb_amr. Run the script from PC (DOS) command prompt.
Note: Please verify the input, output and image path before running the script.

7 Pre compilation Options

7.1 Test application

The following C options need to be set

C Defines	Description	Remarks
WBAMR_BIG_ENDIAN	To run the code as Big Endian	
LITTLE_ENDIAN	To run the code as Little Endian.	
TIME_PROFILE	Profiling	For the board

7.2 Library

C Defines	Description	Remarks
WBAMR_ARM1136	To enable ARM11 optimizations	ELINUX build for ARM11 only
WBAMR_ADS	ARM9 RVDS syntax only	