08-7158-RN-ZCH66
OCT209, 2008
3.1

Release Notes for NB-AMR Decoder and Encoder

ABSTRACT:

Release Notes for NB-AMR Decoder and Encoder

KEYWORDS:

Multimedia codecs, NBAMR, speech

Revision History

VERSION	DATE	AUTHOR	CHANGE DESCRIPTION
1.0	17-Jan-2005	Ashok Kumar	Final release of NB-AMR codec on RVDS
2.0	04-Mar-2005	Ashok Kumar	Final release of NB-AMR codec on RVDS
2.1	15-Sep-2005	Anand	Build procedure for RVDS 2.2
3.0	06-Feb-2006	Lauren Post	Using new format
3.1	08-Oct-2008	jackiea	Update for ARM9

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 ARM11 ELINUX,ARM9 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 Speech Signal processing and NB-AMR vocoder.

1.4 References

1.4.1 Standards

- **ETSI EN 301 703 V7.0.2 (1999-12)** - Digital cellular telecommunications system (Phase 2+) (GSM); Adaptive Multi-Rate (AMR); Speech processing functions; General description (GSM 06.71 version 7.0.2 Release 1998)
- **ETSI EN 301 704 V7.2.1 (2000-04)** - Digital cellular telecommunications system (Phase 2+) (GSM); Adaptive Multi-Rate (AMR) speech transcoding (GSM 06.90 version 7.2.1 Release 1998).
- **ETSI EN 301 705 V7.1.1 (2000-04)** - Digital cellular telecommunications system (Phase 2+); Substitution and muting of lost frames for Adaptive Multi Rate (AMR) speech traffic channels (GSM 06.91 version 7.1.1 Release 1998).
- **ETSI EN 301 706 V7.1.1 (1999-12)** - Digital cellular telecommunication system (Phase 2+); Comfort noise aspects for Adaptive Multi-Rate (AMR) speech traffic channels (GSM 06.92 version 7.1.1 Release 1998)
- **ETSI EN 301 707 V7.3.1 (2001-03)** - Digital cellular telecommunications system (Phase 2+); Discontinuous Transmission (DTX) for Adaptive Multi-Rate (AMR) speech traffic channels (GSM 06.93 version 7.3.1 Release 1998).
- **ETSI EN 301 708 V7.1.1 (1999-12)** - Digital cellular telecommunications system (Phase 2+); Voice Activity Detector (VAD) for Adaptive Multi-Rate (AMR) speech traffic channels; General description (GSM 06.94 version 7.1.1 Release 1998).

- **ETSI EN 301 712 V7.4.1 (2000-09)** - Digital cellular telecommunications system (Phase 2+); Adaptive Multi Rate (AMR) speech; ANSI-C code for the AMR speech codec (GSM 06.73 version 7.4.1 Release 1998).
- **ETSI EN 301 713 V7.0.3 (2000-10)** - Digital cellular telecommunications system (Phase 2+); Test sequences for the Adaptive Multi-Rate (AMR) speech codec (GSM 06.74 version 7.0.3 Release 1998).
- **ITU-T Recommendation G.711 (1988)** – Coding of analogue signals by pulse code modulation Pulse code modulation (PCM) of voice frequencies.
- **3GPP TS 26.101 V5.0.0 (2002-06) -** Adaptive Multi-Rate (AMR) speech codec frame structure.

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

- NB AMR Codec Application Programming Interface – nbamr_codec_api.doc
- NB AMR Codec Requirements Book – nbamr_codec_reqb.doc
- NB AMR Codec Test Plan - nbamr_codec_test_plan.doc
- NB AMR Codec Release notes - nbamr_codec_release_notes.doc
- NB AMR Codec Test Results – nbamr_codec_test_results.doc
- NB AMR Codec Performance Results – nbamr_codec_perf_results.doc
- NB AMR Interface Common Header – nbamr_common_api.h
- NB AMR Interface Decoder Header – nbamr_dec_api.h
- NB AMR Interface Encoder Header – nbamr_enc_api.h
- NB AMR Decoder Application Code – nbamr_dectest.c
- NB AMR Encoder Application Code – nbamr_enctest.c

1.5 Definitions, Acronyms, and Abbreviations

TERM/ACRONYM	DEFINITION
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
ITU	International Telecommunication Union
LSP	Line Spectral Pair
LP	Linear Prediction
MIPS	Million Instructions per Second
NB AMR	Narrow and Adaptive Multi-Rate Codec
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
VAD	Voice Activity Detection

1.6 Document Location

docs/nb_amr

2 Release History

RELEASE NUMBER	DELIVERABLES	FEATURES
1.0		• Engineering Release
2.0	• Interface header file for encoder and decoder • API, ReqB, test plan, test result doc • C source and ASM files • Makefile to generate library	• 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	• More scripts are added to test codec on board
2.4	• 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	• Shared library support • Upgrade to RVDS 2.2 • Bus Alignment Fixes • Name Collision fixes

Table 1. Details of the release

2.1 Assumptions and Known Problems

None

2.2 Contacts

Please report any problems to the following email address: mmsw@freescale.com

3 List of Deliverables

3.1 Documentation

Base directory: / fsl_mad_multimedia_codec /

Directory	Files	Description
docs/nb_amr	nbamr_codec_api.doc nbamr_codec_reqb.doc nbamr_codec_test_plan.doc nbamr_codec_test_results.doc nbamr_codec_perf_results.doc nbamr_codec _release_notes.doc nbamr_codec _datasheet.doc	Application Programming Requirements Book Test Plan Test Results Performance Results Release Notes Datasheet

3.2 Public Headers

Base directory: / fsl_mad_multimedia_codec /

Directory	Files	Description
ghdr	nbamr_common_api.h nbamr_enc_api.h nbamr_dec_api.h	NB_AMR common, encoder and decoder header file

3.3 Test Application Source

Base directory: / fsl_mad_multimedia_codec /

Directory	Files	Description
test /nb_amr	Makefile	makefile to build executables for RVDS, ELINUX and Unix for decoder and encoder
test /nb_amr/c_src	common encoder decoder	Folders containing the c source file for the sample test application
Test/nb_amr/hdr	*.h	Header files for test application
Utils/nb_amr	*.bat *.sh	Batch for running test cases and output comparison on RVDS Scripts for running the test cases and file comparison on the board

3.4 Library Source

Base directory: / fsl_mad_multimedia_codec /

Subdirectory	Files

src/nb_amr	Makefile “Makefile” for building RVDS, UNIX, and ELINUX libraries. lib_nb_amr_dec_arm9_elinux.a: static dec library for MX21 lib_nb_amr_dec_arm9_elinux.so: dynamic dec library for MX21 lib_nb_amr_dec_arm11_elinux.a: static dec library for MX31 lib_nb_amr_dec_arm11_elinux.so: dynamic dec library for MX31 lib_nb_amr_dec_arm9_lervds.a: ARM9 dec LE RVDS library lib_nb_amr_dec_arm11_lervds.a: ARM11 dec LE RVDS library lib_nb_amr_enc_arm9_elinux.a: static enc library for MX21 lib_nb_amr_ enc _arm9_elinux.so: dynamic enc library for MX21 lib_nb_amr_ enc _arm11_elinux.a: static enc library for MX31 lib_nb_amr_ enc _arm11_elinux.so: dynamic enc library for MX31 lib_nb_amr_ enc _arm9_lervds.a: ARM9 enc LE RVDS library lib_nb_amr_ enc _arm11_lervds.a: ARM11 enc LE RVDS library
src/ nb_amr /c_src	*.c, nb_amr source code
src/ nb_amr /hdr	*.h nb_amr library header files

3.5 Common Makefiles

Base Directory: / fsl_mad_multimedia_codec /

Makefile	**Description**
common.mk	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 (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.
common_testapp.mk	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 7. CODEC_LIB generation

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 and ARM926EJ-S). The details for the build procedure are described below.

Note: The build procedure is explained with encoder as an example to build library for the decoder apply the same procedure given below, with the makefile ‘Makefile’.

5.1 Library

To build the library, run ‘make’ from the library 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** : This is the default option and builds both static library ‘lib_nb_amr_enc_arm11_ELINUX.a’ and shared library ‘lib_nb_amr_enc_arm11_ELINUX.so’ , for testing on the board.

- **BUILD=ARM9ELINUX** : This is the default option and builds both static library ‘lib_nb_amr_enc_arm11_ELINUX.a’ and shared library ‘lib_nb_amr_enc_arm11_ELINUX.so’ , for testing on the board.

- **BUILD=ARM11LERVDS**: This option builds the static library ‘lib_nb_amr_enc_arm11_lervds.a’, for testing on RVDS (Armulator).

- **BUILD=ARM9LERVDS**: This option builds the static library ‘lib_nb_amr_enc_arm9_lervds.a’, for testing on RVDS (Armulator).

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 board, RVDS, Linux and Unix. The executables test_nb_amr_enc_arm11_lervds (test_nb_amr_dec_arm11_lervds for decoder), test_nb_amr_enc_arm9_lervds (test_nb_amr_dec_arm9_lervdsfor decoder) for RVDS, test_nb_amr_enc_arm11_elinux (test_nb_amr_dec_arm11_elinux for decoder), test_nb_amr_enc_arm9_elinux (test_nb_amr_dec_arm9_elinuxfor decoder) for board and. 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 option builds the executable 'test_nb_amr_enc_arm11_elinux', for the board.

   - **BUILD=ARM9ELINUX**: This option builds the executable 'test_nb_amr_enc_arm9_elinux', for the board.

   - **BUILD=ARM11LERVDS**: This option builds the executable 'test_nb_amr_enc_arm11_lervds' for the RVDS (Armulator).

   - **BUILD=ARM9LERVDS**: This option builds the executable 'test_nb_amr_enc_arm9_lervds' for the RVDS (Armulator).

2) **LIBRARY options**:
   - **LIB= STATIC**: This option builds the ELINUX test application linked with the ELINUX static library 'lib_nb_amr_enc_arm11_elinux.a'.If nothing is specified ,the executable links with shared library 'lib_nb_amr_enc_arm11_elinux.so'

     Eg: make LIB=STATIC CODER=enc

3) **CODEC options**:
   - **CODER=dec**: This option builds the test application for the NB_AMR decoder. If nothing is specified the encoder test application will be built.
   - **CODER=enc**: This option builds the test application for the NB_AMR encoder. If nothing is specified the encoder test application will be built.

     Eg: make CODER=enc

6 Test Application Execution

6.1 Scripts

In the test/test_util/scripts directory, a script file exists for doing batch processing on several vectors. The script can be modified or parameters set to specify the binaries to use.

6.2 ELINUX

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

a) Go to the directory test_util/scripts" 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/exe to the current directory
d) Copy the required script file (.sh) from test_util/scripts into the working directory on the board
e) Compare output of encoder and decoder using diff script provided in test_util/scripts.

6.3 RVDS

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

6.4 UNIX Reference

The script described in ELINUX execution can be used for C reference. Modify the script to set the executable to be used (nbamr_enc_UNIX for encoder and nbamr_dec_UNIX for decoder).

7 Pre compilation Options

The following C can be set in the test application makefile

C Defines	Description	Remarks
ALIGNMENT_CHECK	This uses co-processor instructions for checking the alignment	
ARM_OPTS	This is used to use extern tables for RVDS/ELINUX	Defined by default
TIME_PROFILE_RVDS	For output of cycles info from RVDS	To be used with RVDS builds only
NBAMR_BIG_ENDIAN	To run the code as Big Endian.	To be used with RVDS builds only
TIME_PROFILE	For output of cycles info from board	To be used with ELINUX builds only. Defined by default