08-6829-RN-ZCH66
MAR 17, 2009
1.0-D01

Release Notes for WMV789 Decoder on ARM11 ELINUX

ABSTRACT:

Release Notes for WMV789 Decoder on ARM11 ELINUX

KEYWORDS:

Multimedia codecs, WMV9, Windows Media Video 9

Revision History

VERSION	DATE	AUTHOR	CHANGE DESCRIPTION
1.0	06-May-2005	Prachi/Anurag	Release notes template
2.0	26-May-2005	Prachi	Release
3.0	16-Sep-2005	Puneet Gulati	Build Procedure changes for RVDS2.2
4.0	06-Feb-2006	Lauren Post	Using new format
4.1	31-Mar-2006	Prachi	Updated
4.2	24-June-2006	Sriram	Updated
4.3	13- Feb- 2007	Abhishek Mehrotra	Updated
4.4	17-Mar-2009	Eagle Zhou	Add support of device mxc_mem

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, 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 video processing and windows media video coding standard.

1.4 References

1.4.1 Standards

- WMV Version 9.0, Windows Media Video V9 Decoding Specification, revision 87
- WMT Version 9.0, Functional Specification, Recommended Media Decoding – rev 8.1.

1.4.2 References

- Arm codec coding guidelines
- Advanced System format (ASF) Specification, Revision 01.20.02, Microsoft Corporation, June 2004

1.4.3 Freescale Multimedia References

- WMV789 Decoder Requirements Book – wmv789_dec_reqb.doc
- WMV789 Decoder Test Plan – wmv789_dec_test_plan.doc
- WMV789 Decoder Release notes – wmv789_dec_release_notes.doc
- WMV789 Decoder Test Results – wmv789_dec_test_results.doc
- WMV789 Decoder Interface Header – wmv789_dec_api.h
- WMV789 Decoder Application Code – wmv789_testapp.c

1.5 Definitions, Acronyms, and Abbreviations

TERM/ACRONYM	DEFINITION
AVC	Advanced Video Coding
API	Application Programming Interface
ARM	Advanced RISC Machine
ASF	Advanced System Format,
FSL	Freescale
ISO	International Standards Organization
ITU	International Telecommunication Union
MPEG	Moving Pictures Expert Group
NAL	Network Abstraction Layer
RVDS	RealView Development Suite
SP	Simple Profile
RVDS	ARM RealView Development Suite
TBD	To Be Determined
UNIX	Linux PC x/86 C-reference binaries
WMV	Windows Media Video

1.6 Document Location

docs/wmv789_dec

2 Release History

RELEASE NUMBER	DELIVERABLES	FEATURES
1.0		• Engineering Release
2.0	• Documentation • Application Interface header file • ELINUX and RVDS libraries and test applications • UNIX/Linux x86 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	• Reference version of the library using only C source. • Assembly optimized code for ARM11. • Addition of debug logs. • Support to build decoder application for board. • Enhanced Application to display the decoded frames in LCD (SW color conversion) •
2.2	Same	• Shared Library Support • Upgrade to RVDS 2.2
4.4	Eagle Zhou	Mxc_mem device support(please get more detail from BSP)

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: /ARM11/

Subdirectory	Files
docs/WMV789_dec	wmv789_dec_api.doc wmv789_dec_reqb.doc wmv789_dec_test_plan.doc wmv789_dec_test_results.doc wmv789_dec_perf_results.doc wmv789_dec _release_notes.doc

3.2 Public Headers

Base directory: /ARM11/

Subdirectory	File
API_include	wmv789_dec_api.h

3.3 Test Application Source

Base directory: /ARM11/src/Video/WMV789_dec

Subdirectory	Files
test/	"Makefile" makefile for building RVDS, UNIX and ELINUX board executables.
test/c_source	*.c, application code.
test/include	*.h, application header files
test/test_util/scripts	scripts for testing

3.4 Library Source

Base directory: /ARM11/src/Video/WMV789_dec

Subdirectory	Files
library	Makefile "Makefile" for building RVDS, UNIX, and ELINUX libraries. libwmv789_dec_arm11_RVDS.a – Special options for simulator testing libwmv789_dec_arm11_ELINUX.a - static library for board libwmv789_dec_arm11_ELINUX.so – shared library for board libwmv789_dec_UNIX.a – library for Linux x/86 – c reference code
library/c_source	*.c, WMV789 decoder source code

library/asm_source	*.s WMV789 decoder assembly source
library/include	*.h, WMV789 decoder library header files

3.5 Common Makefiles

Base Directory: /ARM11/common

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

3.6 Test Vectors

Base Directory: multimedia_vectors/test_vectors

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

Subdirectory	Files
WMV789_dec/input	Input test vectors
WMV789_dec/ref	Reference vectors to bitmatch against

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 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**:

BUILD= ELINUX : This is the default option and builds both static library 'libwmv789_dec_arm11_ELINUX.a' and shared library 'libwmv789_dec_arm11_ELINUX.so', for testing on the board.

Note: make sure that WMV9_SIMPLE_ONLY=1 is defined when making with this option.

BUILD=RVDS: This option builds the static library 'libwmv789_dec_arm11_RVDS.a', for testing on RVDS (Armulator).

BUILD=UNIX: This option builds the static library 'libwmv789_dec_UNIX.a', for testing on UNIX/Linux machine.

Eg:
```
make BUILD= ELINUX WMV9_SIMPLE_ONLY=1
make BUILD=RVDS
make BUILD=UNIX
```

b) **ENDIAN options for RVDS**:

- **TARGET_ENDIAN=LITTLE**: This is the default option and sets the endian-ness to 'little'
- **TARGET_ENDIAN=BIG**: This option sets the endian-ness to big
  Eg: make BUILD=RVDS TARGET_ENDIAN=BIG

c) **REF_CODE=1:** This option is used when Microsoft API's are used. When this option is not enabled then standard API's are used.

d) **clean options**:

- **clean_RVDS**: Deletes all the object files and the RVDS library 'libwmv789_dec_arm11_RVDS.a'.
- **clean_ELINUX**: Deletes all the object file and the ELINUX libraries libwmv789_dec_arm11_ELINUX.a and libwmv789_dec_arm11_ELINUX.so.
- **clean_UNIX**: Deletes all the object files and the UNIX library 'libwmv789_dec_UNIX.a'.
- **clean**: Deletes all the object files and RVDS,UNIX and ELINUX libraries.

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

The library that is built is saved as libwmv789_dec_arm11_RVDS.a for RVDS build, and libwmv789_dec_arm11_ELINUX.a and libwmv789_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	PC (Using Cygwin)	BUILD= ELINUX WMV9_SIMPLE_ONLY=1	libwmv789_dec_arm11_ELINUX.a libwmv789_dec_arm11_ELINUX.so
RVDS	PC (Using Cygwin)	BUILD=RVDS WMV9_SIMPLE_ONLY=1 TARGET_ENDIAN=BIG/ LITTLE	libwmv789_dec_arm11_RVDS.a
Unix/ Linux	Linux/ Unix machine	BUILD=UNIX TARGET_ENDIAN=BIG/ LITTLE	libwmv789_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 wmv789_dec_arm11_RVDS for RVDS, wmv789_dec_arm11_ELINUX for board and wmv789_dec_UNIX for UNIX are stored under the 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=ELINUX**: This is the default option and builds the executable 'wmv789_dec_arm11_ELINUX', for the board.
   - **BUILD=RVDS**: This option builds the executable 'wmv789_dec_arm11_RVDS' for the RVDS (Armulator).
   - **BUILD=UNIX**: This option builds the executable 'wmv789_dec_UNIX' for the Unix/Linux machine.

   Eg: make BUILD=ELINUX (for board)
   make BUILD=RVDS (for Armulator)

make BUILD=UNIX (for Unix/Linux machine)

2) **ENDIAN options for RVDS**:
   - **TARGET_ENDIAN=LITTLE**: This is the default option and sets the endian-ness to 'little'

   - **TARGET_ENDIAN=BIG**: This option sets the endian-ness to big
     Eg: make BUILD=RVDS TARGET_ENDIAN=BIG

3) **LIBRARY options**:
   - **LIB= STATIC**: This option builds the ELINUX test application linked with the ELINUX static library 'libwmv789_dec_arm11_ELINUX.a'.If nothing is specified, the executable links with shared library 'libwmv789_dec_arm11_ELINUX.so'
     Eg: make LIB=STATIC
   - **REF_CODE=1:** This option is used when Microsoft API's are used. When this option is not enabled then standard API's are used.

4) **clean options**:
   - **clean_RVDS**: Deletes all the object files and the RVDS executable 'wmv789_dec_arm11_RVDS' .
   - **clean_ELINUX**: Deletes all the object file and the ELINUX 'wmv789_dec_arm11_ELINUX'.
   - **clean_UNIX**: Deletes all the object files and the Unix/Linux executable 'wmv789_dec_UNIX'.
   - **clean**: Deletes all the object files and RVDS,UNIX ELINUX executables.

5) **clean options**:
   - USE_DISPLAY=1 Builds the application with display capabilities. If built without this option, display would not be enabled. This is supported only for board.
   - NO_KEV=1 This is used when output is not required. By default, kev is generated.
   - TIME_PROFILE=1 Builds the application with no display and output capabilities. This eliminates the overhead due to file writes and color conversion routines. Enabling this switch would automatically enable NO_KEV. This is a useful build to gauge the libraries performance towards a given bitstream. Please note that other application overheads like bitstream read and other calling overheads still remain. This is only to be used for build.

Note:
In 'common_testapp.mk' 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 'common_testapp.mk' 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=ELINUX LIB= STATIC	wmv789_dec_arm11_ELINUX
RVDS	PC (Using Cygwin)	BUILD=RVDS TARGET_ENDIAN=LITTLE/BIG	wmv789_dec_arm11_RVDS

UNIX/ Linux	Unix/Linux machine	BUILD=UNIX TARGET_ENDIAN=LITTLE/BIG	wmv789_dec_UNIX

6 Test Application Execution

To know the options provided by the test application, run the executable without any argument. It shall print a brief summary of all the options available.

a. -in <file_name> :input file name
b. –o <out fdir> :Output directory
c. –e <num frames> : numbers of frames to be decoded

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

wmv789_dec_arm11_ELINUX –i <input vector> -o <output vector>

The output vector will be placed into file <output_vector> which can include a path

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 UNIX Reference

To execute on Linux x/86 type:

wmv789_dec_arm11_UNIX –i <input vector> -o <output vector>

7 Pre compilation Options

The following C options need to be set

C Defines	Description	Remarks
ADS	To run on RVDS. Disable to run on UNIX.	
ALL_PROFILE	doesn't use WMV9_SIMPLE_ONLY	
ARM11		defined for all builds
LITTLE_ENDIAN	To run the code as Little Endian.	
LOG_TIMING	Performance logging	
MXC	Used for display	
SAVE_KEV	Set when NO_KEV is not set	Default case – used for ELINUX builds only
USE_ASM		Used for all builds
USE_DISPLAY	Displays video output	
WMV9_SIMPLE_ONLY	Set when `ALL_PROFILE is not set`	