



09-7429-RN-ZCH70

13-MAR-2009

1.0

Release Notes for MPEG4-ASP Decoder on ARM9 ELINUX

ABSTRACT:

Release Notes for MPEG4-ASP Decoder on ARM9 ELINUX

KEYWORDS:

Multimedia codecs, MPEG4

APPROVED:

Wang Zening

Revision History

VERSION	DATE	AUTHOR	CHANGE DESCRIPTION
1.0	13-MAR-2009	B07235	Creation

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 Freescale Multimedia References	4
1.5 Definitions, Acronyms, and Abbreviations	5
1.6 Document Location	5
2 Release History	6
2.1 Assumptions and Known Problems	6
2.2 Contacts	6
3 List of Deliverables.....	7
3.1 Documentation	7
3.2 Public Headers.....	7
3.3 Release files.....	7
3.4 Test Application Source	7
3.5 Library Source.....	7
3.6 Common Makefiles	8
4 Software Setup & Tools used	9
5 Build Procedure.....	10
5.1 Library.....	10
5.2 Test Application	11
6 Test Application Execution	13
Usages of test application.....	13
6.1.....	13
6.2 Typical commands for test application	13
7 Pre compilation Options	15

Introduction

1.1 Purpose

The purpose of this document is to provide the followings:

- Information on the package contents;
- Instructions on building library and wrapper for test applications;
- Test execution on ARM9 ELINUX.

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.

1.3 Audience Description

The reader is expected to have basic understanding of video video coding standard.

1.4 References

1.4.1 Standards

- ISO/IEC 14496-2:2003 Information technology -- Coding of Audio-Visual Objects – Part2: Visual
- ITU-T H.263 video coding specification.
- ITU-T H.263 Annex X, Profiles and levels definition (SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS, Infrastructure of audiovisual services – Coding of moving video, 4/2001)

1.4.2 Freescale Multimedia References

- MPEG4 Decoder Application Programming Interface – mpeg4_asp_dec_api.doc
- MPEG4 Decoder Release Notes – mpeg4_asp_dec_release_notes.doc
- MPEG4 Decoder Data Sheet – mpeg4_asp_dec_datasheet.doc
- MPEG4 Interface Header – mpeg4_asp_api.h
- MPEG4 Application code – dut_mpeg4_asp_api_vts.c

1.5 Definitions, Acronyms, and Abbreviations

TERM/ACRONYM	DEFINITION
API	Application Programming Interface
ARM	Advanced RISC Machine
FSL	Freescale
ISO	International Standards Organization
ITU	International Telecommunication Union
MPEG	Moving Pictures Expert Group
TBD	To Be Determined
ASP	Advanced Simple Profile

1.6 Document Location

docs/MPEG4ASP_dec

2 Release History

RELEASE NUMBER	DELIVERABLES	FEATURES
1.0	<ul style="list-style-type: none"> • Documentation • Application Interface header file • ELINUX libraries and the test application • Makefiles and Source code for library and test application including optimized assembler for the ELINUX libraries. 	<ul style="list-style-type: none"> • Assembly optimized code for ARM9. • Enhanced Application to display the decoded frames in LCD • Fully ASP Support

Table 1. Details of the Release

2.1 Assumptions and Known Problems

Know Issues	Description
ENGR00108323	In the current design, the temp buffer for intra row's IDCT is 16bits, it maybe bring overflow for special streams(mat045_reva.m4v).

2.2 Contacts

Please report any problems, contact Freescale customer representative.

3 List of Deliverables

There are 2 ways of package releases. Whether the package contains the source code of library depends on the license.

You can get this deliverable package from the following link:

<http://compass.freescale.net/go/202542123>

3.1 Documentation

Base directory: /docs

Subdirectory	Files
/MPEG4ASP_dec	mpeg4_asp_dec_api.doc mpeg4_asp_dec_datasheet.doc mpeg4_dec_release_notes.doc

3.2 Public Headers

Base directory: /

Subdirectory	File
ghdr	mpeg4_asp_api.h

3.3 Release files

Base directory: /release/

Subdirectory	Files
/lib	lib_MPEG4ASP_dec_arm9_elinux.so It is the dynamic library for arm9 platform.
/exe	test_dec_arm_elinux is the executable the run the test application.

3.4 Test Application Source

Base directory: /test/video_test_vts20

Subdirectory	Files
/mpeg4_asp_dut/	“Makefile” makefile for building ELINUX board executables.
/mpeg4_asp_dut/src	*.c, application code.
/mpeg4_asp_dut/hdr	*.h, application header files
/dut_hdr	*.h, vts api header files

3.5 Library Source

Base directory: /src/MPEG4ASP_dec

Subdirectory	Files
/	Makefile “Makefile” for building ELINUX libraries.
/c_src	*.c, mpeg4 decoder source code
/arm_asm	*.s mpeg4 decoder assembly source
/hdr	*.h, mpeg4 decoder library header files

3.6 Common Makefiles

Base Directory: / fsl_mad_multimedia_codec/build

Makefile	Description
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 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. Codec header path 13. 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

4 Software Setup & Tools used

- ARM RVDS 2.2 should be installed in the PC.
- Freescale Linux OS Release: imx27_ads_20071219-rel3-1tib.iso (Bono, MX27). You can download the bsp from www.freescale.com
http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=i.MX27&nodeId=0162468rH311432973ZrDR&fosp=1&tab=Design_Tools_Tab .
- When building the BSP, please configure the toolchain : gcc-4.1.2-glibc-2.5-nptl-3
- ‘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.
- Update the toolchain path, using local toolchain path replace the default in the “build/Makefile.init” and “build/Makefile_test.init”. Specifically, user should update the variable TOOLS_DIR, LINK_TC_ROOT by user’s local path. When build the test application, the library path should be updated.

5 Build Procedure

All the required makefiles are provided under individual directories. The library can be built for target processor MX27(ARM926EJ-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 following options are available to build the library.

Options

a) BUILD options:

BUILD= ARM9ELINUX : This is the default option and builds both static library ‘lib_MPEG4ASP_dec_arm9_elinux.a’ and shared library ‘lib_MPEG4ASP_dec_arm9_elinux.so’, for testing on the board.

Eg: make BUILD= ARM9ELINUX

b) clean options:

- **clean:** Deletes all the object files and libraries.

Note: Make appropriate changes in file ‘Makefile.init’ at directory ‘build’ for the location of toolchains.

The library that is built is saved as lib_MPEG4ASP_dec_arm9_elinux.a and lib_MPEG4ASP_dec_arm9_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= ARM9ELINUX	lib_MPEG4ASP_dec_arm9_elinux.a lib_MPEG4ASP_dec_arm9_elinux.so

5.2 Test Application

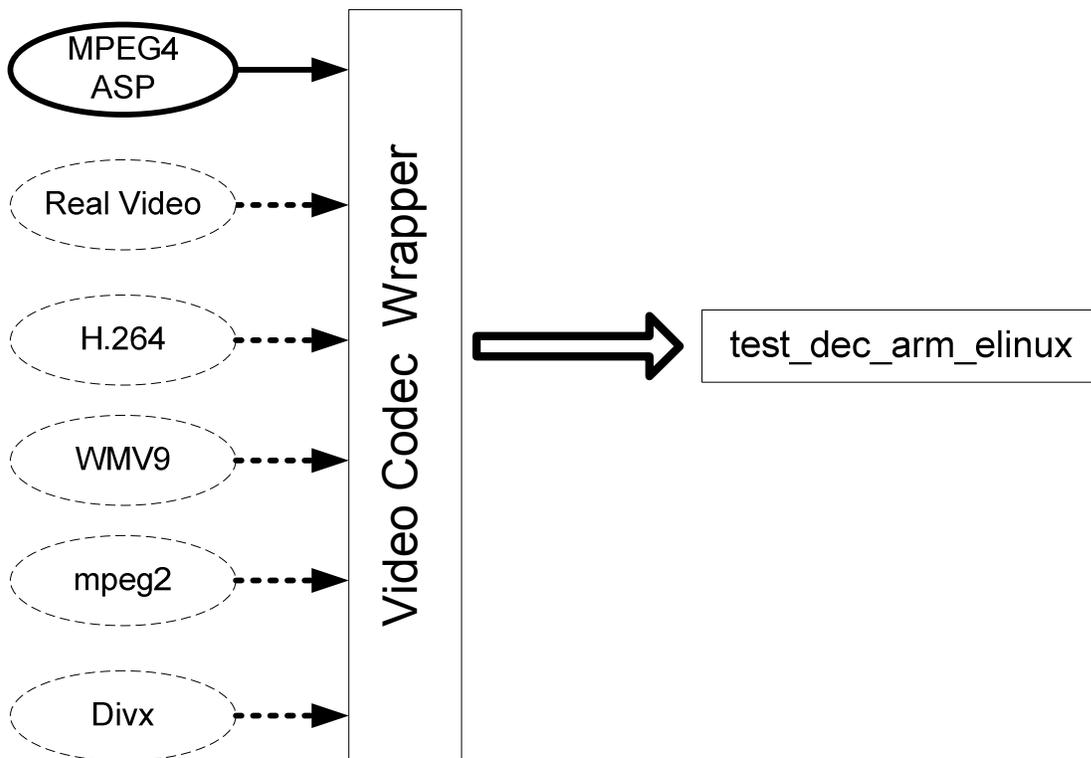


Figure 1 test application Architecture

As figure1 shows, to unify the test application interfaces for all video decoders, the video codec wrapper layer was added.

- Ellipse represent video codecs' library
- VCW(Video Codec Wrapper) represents the unified interfaces layer
- test_dec_arm_elinux is the unified test application execution, users can get it from the directory.

Build Wrapper Library

To build the test application, run 'make' on 'Makefile' from the mpeg4_asp_dut directory. This makefile can create a wrapper library for testing on board for ARM9. The shared library libDutDecMPEG4ASP-2.1.so for board is stored under the test/video_test_vts20/mpeg4_asp_dut directory. The makefile shall create the required directory structure to hold the object files and shared library. The following commands should be invoked so as to build the executables.

Options

1) BUILD options:

- **BUILD=ARM9ELINUX:** This is the default option and builds the shared library 'libDutDecMPEG4ASP-2.1.so', for the board.

Eg: make BUILD=ARM9ELINUX (for board)

2) **LIBRARY options:**

- **LIB_TYPE= STATIC:** This option builds the ELINUX test application linked with the ELINUX static library 'lib_MPEG4ASP_dec_arm9_elinux.a'. If nothing is specified, the executable links with shared library 'lib_MPEG4ASP_dec_arm9_elinux.so'

Eg: make LIB_TYPE=STATIC

3) **clean options:**

- **clean:** Deletes all the object files and executables.

The following table summarises the build options,

Target	Compilation Environment	Build options	Shared library
Board	Redhat Linux Machine	BUILD=ARM9ELINUX LIB_TYPE= STATIC	libDutDecMPEG4ASP-2.1.so

6 Test Application Execution

The wrapper library, libDutDecMPEG4ASP-2.1.so, should run with the executable test_dec_arm_elinux provided in the directory /release.

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.

6.1 Usages of test application

The tester application usage is described below:

Usage:

```
./test_dec_arm_elinux [options] -l dynamic_library -i bitstream_file
```

options:

```
-o <file_name>      : Save decoded output in YUV 4:2:0 format
                    [default: no save]
-n <frame_num>      : decode max <frame_num> frames
                    [default: all frames will be decoded]
-t <frame time log> : if specified, produce every frame decoding
                    time in log file.
-r <report file>    : if specified, produce test information in
                    report_file.
-d                  : if specified, LCD render enabled.
-m                  : if specified, print stack and heap infor.
-w, [wrapper options] : if specified, pass options to wrapper.
-v                  : if specified, print library version.
```

For wrapper options, users can get the usage by the following instruction:

```
./test_dec_arm_elinux -l libDutDecRealVideo-1.8.so -i "test_vector_name" -w,-h
```

```
Usage: MPEG4 ASP Dut Wrapper [options]
```

```
-h          Display this command line help and exit
-s=<num>    Enable skip non-key frames [default is off]
  0         Only skip B frames
  1         Skip B&P frames
```

6.2 Typical commands for test application

Typical command usage without display:

Normal command:

```
./test_dec_arm_elinux -i <input vectors path/bitstream_file> -l <wrapper
library>
```

Enable skip B frames:

```
./test_dec_arm_elinux -i <input vectors path/bitstream_file> -l <wrapper  
library>, -w,-s=0
```

Enable skip B&P frames:

```
./test_dec_arm_elinux -i <input vectors path/bitstream_file> -l <wrapper  
library>, -w,-s=1
```

7 Pre compilation Options

The following C options need to be set

C Defines	Description	Remarks
ARM9_C_VERSION	Use C code	Used for all builds
ARM_ASM	Use arm asm	Used for all builds
ARMv6ASM	Use arm v6 asm	Used for ARMv6 build
ARMv5ASM	Use arm v5 asm	Used for ARMv5 build