



08-6272-RN-ZCH66

MAR. 11, 2009

6.1

Release Notes for H.264/AVC Decoder

ABSTRACT:

Release Notes for H.264/AVC Decoder on ARM11 ELINUX , ARM11 WINCE, ARM9 ELINUX, ARM9 WINCE, X86 LINUX, X86 Windows

KEYWORDS:

Multimedia codecs, H.264, MPEG

Approved:

Wang Zening

Revision History

VERSION	DATE	AUTHOR	CHANGE DESCRIPTION
0.1	07-May-2004	Murali/ Pradeep	Release notes template
0.2	11-May-2004	Murali/ Pradeep	Review rework
1.0	22-Dec-2004	Chandra	Formatted for release 1.0
2.0	14-Apr-2005	Chandra	Updated with changes made for SCMA11 board testing
2.1	09-Sep-2005	Sukruth	RVDS2.2 build procedure changes
3.0	11-Nov-2005	Anurag	Modified for IPU Integration with H264 Decoder
3.1	07-Dec-2005	Madhu	Makefile changes for default hardware IPU enabled
4.0	06-Feb-2006	Lauren Post	Using new format
4.1	01-Apr-2006	Manoj Arvind	Updated makefile options and directory structure.
4.2	09-Jun-2006	Dheeraj C Kuchangi	Updated makefile options.
4.3	21-Sep-2006	Vineet Golchha	Extra CFLAG added in test application makefile
4.4	26-Sep-2006	Vineet Golchha	Updated document for VRTX
4.5	27-Sep-2006	Manoj Arvind	Update common make options
4.6	28-Sep-2006	Manoj Arvind	Included Review comments
4.7	12-Jul-2007	Durgaprasad Bilagi	Updated run procedure & release package
5.0	19-Nov-2007	Wang Zening	Update for MAD release
6.0	08-Dec-2008	Chen Zhenyong	Add other platforms support
6.1	11-Mar-2009	Eagle Zhou	Add support of device mxc_mem

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	7
2.2 Contacts	8
3 List of Deliverables	9
3.1 Documentation	9
3.2 Public Headers	9
3.3 Test Application Source	9
3.4 Library Source	9
3.5 Common Makefiles	10
3.6 Files for IDE buildings	10
3.7 Toolchain bug fixing	10
4 Software Setup & Tools used	11
5 Build Procedure	12
5.1 Library	12
5.2 Test Application	13
5.2.1 Build wrapper	13
5.2.2 Build unified application	14
6 Test Application Execution	15

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, ARM11 WINCE, ARM9 ELINUX, ARM9 WINCE, X86 LINUX, X86 Windows.

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 video coding standards.

1.4 References

1.4.1 Standards

1. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), JVT-G050r1.doc
2. Proposed Draft AVC|H.264 Conformance Spec, JVT-J011.doc

1.4.2 Freescale Multimedia References

3. H.264 Decoder Application Programming Interface – h264_dec_api.doc
4. H.264 Decoder Requirements Book - h264_dec_reqb.doc
5. H.264 Decoder Test Plan - h264_dec_test_plan.doc
6. H.264 Decoder Release notes - h264_dec_release_notes.doc
7. H.264 Decoder Test Results – h264_dec_test_results.doc
8. H.264 Decoder Performance Results – h264_dec_perf_results.doc
9. H.264 Decoder Interface Header – avcd_dec_api.h
10. H.264 Decoder Application Code – decoder.c

1.5 Definitions, Acronyms, and Abbreviations

TERM/ACRONYM	DEFINITION
AVC	Advanced Video Coding
API	Application Programming Interface
ARM	Advanced RISC Machine
FSL	Freescale
ISO	International Standards Organization
IPU	Image Processing Unit
ITU	International Telecommunication Union
MPEG	Moving Pictures Expert Group
NAL	Network Abstraction Layer
PAF	Parallelization Across Frames
RVDS	ARM RealView Development Suite
TBD	To Be Determined
UNIX	Linux PC x/86 C-reference binaries

1.6 Document Location

docs/h264_dec

2 Release History

RELEASE NUMBER	DELIVERABLES	FEATURES
1.0	<ul style="list-style-type: none"> The 'C' source and header files Assembly code for ARM11. 'makefile' to build the library and test application. The test application source code. The input test bit streams and reference output files (kev format). 	<ul style="list-style-type: none"> Reference version of the library using only C source. Assembly optimized code for ARM11. Addition of debug logs.
2.0	All deliverables as listed in release 1.0 plus <ul style="list-style-type: none"> Common makefile to build library for RVDS and board. Two 'makefiles' for the application, one to build application for RVDS simulations and the other to build the application for the board. A common 'makefile' for the application that supports builds for other platforms (e.g. UNIX). 	<ul style="list-style-type: none"> Support to build decoder application for board. Enhanced Application to display the decoded frames in LCD (SW color conversion)
3.0	All deliverables as listed in previous releases plus <ul style="list-style-type: none"> Common makefile to build library for RVDS, board & board with IPU. Two 'makefiles' are provided, one to build application for RVDS simulations and the other to build the application for the board / board with IPU. A common 'makefile' for the application that supports builds for other platforms (e.g. UNIX). 	<ul style="list-style-type: none"> IPU API's for H264 Decoder IPU Deblocking Integration – H264 Decoder library has been integrated with Hardware Deblocking Filter i.e. IPU for board. Support to build the application with Decoder and IPU API Library has been provided.
3.1	<ul style="list-style-type: none"> Documentation Application Interface header ELINUX and RVDS libraries and test applications UNIX/Linux x/86 Reference library and test application IPU ELINUX libraries and test application Makefiles and Source code for library and test application including optimized assembler for the ELINUX and RVDS libraries. Test vectors 	<ul style="list-style-type: none"> Default hardware IPU deblocking enabled libraries. Shared Library support Upgrade to RVDS 2.2

3.2	<ul style="list-style-type: none"> • All deliverables as listed in previous release 	<ul style="list-style-type: none"> • Parallelization of decoding on ARM with deblocking on IPU for improved performance. • Runtime support for software deblocking for video sequences with height or width greater than 1024 pixels
4.4	<ul style="list-style-type: none"> • All deliverables as listed in previous release 	<ul style="list-style-type: none"> • Implementation of DPB • Additional APIs to support different output formats
4.5	<ul style="list-style-type: none"> • All deliverables as listed in previous release 	<ul style="list-style-type: none"> • Support for ARM11LERVDS is not present in the release BLN_MMCODECS_M AIN_00.12.00
5.0	<ul style="list-style-type: none"> • Documentation • Application Interface header file • ELINUX libraries and test applications • Makefiles and Source code for library and test application including optimized assembler for the ELINUX libraries. 	<ul style="list-style-type: none"> • Optimized library for Arm11 • Improved Error Handling/protection • Removed the call back interfaces for input. •
6.0	<ul style="list-style-type: none"> • All deliverables as listed in previous release • Add buildings for x86 linux and windows 	<ul style="list-style-type: none"> •
6.1	<ul style="list-style-type: none"> • All deliverables as listed in previous release 	<ul style="list-style-type: none"> • Mxc_mem device support(please get more detail from BSP)

Table 1. Details of the release

2.1 Assumptions and Known Problems

- The BSP has to support non-4 bytes aligned memory access
- The PF has to support logical address input

2.2 Contacts

Please report any problems, contact Freescale customer representative.

3 List of Deliverables

Note: '*' marked contents are only available for authorized audiences.

3.1 Documentation

Directory	Files
docs/h264_dec	h264_dec_api.doc h264_dec_reqb.doc* h264_dec_test_plan.doc* h264_dec_test_results.doc* h264_dec_perf_results.doc* h264_dec_release_notes.doc H264Decoder_DataSheet.pdf

3.2 Public Headers

Directory	File
ghdr/	avcd_dec_api.h

3.3 Test Application Source

Directory	Files
test/video_test/h264_dut	Wrapper code for this H264 decoder
test/dut_hdr	Unified wrapper APIs
test/uniform_test	Unified test application

3.4 Library Source

Directory	Files
src/H264_dec*	Makefile "MakefileEx" for building ELINUX libraries and x86 linux libraries
src/H264_dec /c_src*	*.c, H264 decoder source code
src/H264_dec /asm_arm*	*.s H264 assembly source
src/H264_dec /c_src/ipu*	*.c, IPU source files
src/H264_dec/hdr/ipu*	*.h, IPU headers
src/H264_dec /hdr*	*.h, H264 decoder library header files

3.5 Common Makefiles

Base Directory: /build/

Makefile	Description
Makefile.toolchain.init	<p>This is a common makefile included in the codec library makefile and test application Makefile. It defines toolchain settings for different platforms, and defines common compile rules.</p> <ol style="list-style-type: none"> 1. Path to toolchain tools (TOOL_CHAIN) 2. GNU header file path (SYS_INCLUDE) 3. GNU library path (LIB_PATHS) 4. GNU Compiler/Assembler Options (CFLAGS, 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. RVDS Compilation Tools 12. Codec header path 13. ELINUXDLIB builds for libraries that must be linked using the toolchain because of external library includes.

3.6 Files for IDE buildings

Windows CE and Windows XP buildings are located under fsl_mm_wince and fsl_mm_winxp respectively. They are built in Microsoft Visual Studio 2003 IDE environment, not using “make”.

3.7 Toolchain bug fixing

When using DVTK building shared library, some section type may be set wrong. The test application cannot load the shared library at run time and report following error:

can't load libDutDecH264.so: ./libDutDecH264.so: unexpected reloc type 0x03.

If this happen, please use “RelocTypeFix” tool to fix it. This tool can be found under utils/DVTK_bug_fixing/RelocTypeFix.

4 Software Setup & Tools used

- ARM RVDS 2.2 (build 503) should be installed in the PC.
- Freescale Linux OS Release L26.18.5 must be running on the evaluation board. The kernel needs to be rebuilt to support this option.
- After the Board is up : Run Command
echo 1 1 1 > /proc/sys/vm/lowmem_reserve_ratio used
- 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 src/H264_dec directory. The makefile can be used if you want to build the library only. The same makefile can be used to build libraries for Unix, Linux (board) and VRTX with different build options. The following options are available to build the library.

Options (for target linux family platforms)

1. BUILD options:

- **BUILD= ARM11ELINUX** : This option builds static library ‘lib_H264_dec_arm11_elinux.a’ and generates object files. To build the shared library, use the Build option BUILD=ARM11ELINUX BIN_TYPE=DLL (without clean) on the linux machine after building the static library on CYGWIN.
- **BUILD= UNIX** : This option builds static library ‘lib_H264_dec_x86_unix.a’ and generates object files. To build the shared library, use the Build option BUILD=UNIX BIN_TYPE=DLL (without clean) on the linux machine after building the static library on CYGWIN.

2. ENDIAN options:

- **TGT_ENDIAN=LITTLE**: This is the default option and sets the endian-ness to ‘little’
- **TGT_ENDIAN=BIG**: This option sets the endian-ness to big

3. clean options:

- **BUILD=ARM11ELINUX clean**: Deletes all the object files and ELINUX libraries.

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

The library that is built is saved as libh264_dec_arm11_elinux.a and libh264_dec_arm11_elinux.so for ELINUX board build. These libraries are saved in the current directory (the same directory in which the source and assembly directories are listed). While making the shared library, it will be copied into the test/H264_dec directory.

Target	Compilation Environment	Build Options	Library Name
i.MX31 Board	PC (Using Cygwin)	BUILD=ARM11ELINUX	libh264_dec_arm11_elinux.a
i.MX31 Board	Using Linux Machine	BUILD=ARM11ELINUX BIN_TYPE=DLL	libh264_dec_arm11_elinux.so

	with gcc family toolchain		
--	------------------------------	--	--

For Windows CE building, Microsoft Platform Builder 5.0 or Microsoft Visual Studio 2003 should be installed, and latest i.MX31 bsp should be setup. Add fsl_mm_wince\Multimedia\core_libs\codec\h264_dec_arm11\dirs to your project and build.

For Windows XP building, Microsoft Visual Studio 2003 should be installed. Open project file fsl_mm_winxp\Multimedia\core_libs\codec\H264DecStaticLib\H264DecStaticLib.vcproj and build.

5.2 Test Application

Test application consists two parts: shared library which wraps H264 decoder with unified APIs; unified test application which can test different decoders.

5.2.1 Build wrapper

5.2.1.1 Target linux family platforms

Run `make -f MakefileEx -C test/video_test/h264_dut/` and get following help message:

Options:

`BUILD={ ARM11ELINUX,ARM11LERVDS,UNIX }` Default to ARM11ELINUX

`RVDS_VERSION={ 2_2,3_0,3_1 }` Default to 2_2 (2.2). This is used when RVDS is selected for building

`TIME_PROFILE_ENABLE={ 0,1 }` Whether time profiling enabled. Default to 1

`RELEASE={ 0,1 }` 0 - Debug version; 1 - Release version. Default to 1

`OPT_LEVEL={ 0,1,2 }` 0 - c; 1 - Arm11 asm; 2 - Neon asm. Default to 0

`BIN_TYPE={ DLL,LIB,EXE }` Default to LIB. Here EXE is forbidden

Select options for your platform. For example, following will make a build for i.MX31+elinux:
`make -f MakefileEx -C test/video_test/h264_dut/ BUILD=ARM11ELINUX BIN_TYPE=DLL all`

To build x86+linux:

`make -f MakefileEx -C test/video_test/h264_dut/ BUILD=UNIX BIN_TYPE=DLL all`

5.2.1.2 Windows CE

Add fsl_mm_wince\Multimedia\core_libs\test\video_test\h264_dut\dirs to your PB project and build.

5.2.1.3 Windows XP

Open project fsl_mm_winxp\Multimedia\core_libs\test\H264DecDutDll\H264DecDutDll.vcproj and build.

5.2.2 Build unified application

5.2.2.1 Target linux family platforms

Run `make -f MakefileEx -C test/video_test/uniform_test/` and get following help message:

`make [Options] all`

Options:

`BUILD={ARM11ELINUX,ARM11LERVDS,UNIX}` Default to ARM11ELINUX

`RVDS_VERSION={2_2,3_0,3_1}` Default to 2_2 (2.2). This is used when RVDS is selected for building

`TIME_PROFILE_ENABLE={0,1}` Whether time profiling enabled. Default to 1

`RELEASE={0,1}` 0 - Debug version; 1 - Release version. Default to 1

`OPT_LEVEL={0,1,2}` 0 - c; 1 - Arm11 asm; 2 - Neon asm. Default to 0

`BIN_TYPE={DLL,LIB,EXE}` Default to EXE. Here DLL and LIB are forbidden

`USE_DLL=1` Link with shared library when specified; otherwise link with static library

Special note: if `USE_DLL` is not set 1, then static link is enabled. In this case, a codec library should be specified by setting following macros in your command line:

`ADDITIONAL_LIBRARY=<path_name>` Provide special video codec static library full path and name

`X11_ENABLE={0,1}` For `BUILD=UNIX`; if enabled, playback window is supported

Select options for your platform. For example, following will make a build for i.MX31+elinux:

`make -f MakefileEx -C test/video_test/uniform_test/ BUILD= ARM11ELINUX USE_DLL=1 all`

To build x86+linux:

`make -f MakefileEx -C test/video_test/uniform_test/ BUILD=UNIX USE_DLL=1 all`

Note: for x86+linux, the test application supports XWindow display. If garget linux supports windows display, it is recommended to add option `X11_ENABLE=1`.

5.2.2.2 Windows CE

Add `fsl_mm_wince\Multimedia\core_libs\test\video_test\ uniform_test\dirs` to your PB project and build.

5.2.2.3 Windows XP

Open project `fsl_mm_winxp\Multimedia\core_libs\test\ DecTest\ DecTest.vcproj` and build.

Note:

In 'Makefile.toolchain.init' at directory '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.toolchain.init' should be modified to point to the directories where the linking and compilation tools are present before building the application for board.

6 Test Application Execution

Run the application following options:

Usage: [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 information.
-w,[wrapper options]	: if specified, pass options to wrapper.
-v	: if specified, print library version.