ARC® Video Subsystem MPEG 2/4 Decoder

MPEG-2 MP@ML, MPEG-4 SP, and MPEG-4 ASP Decoder and MJPEG Decoder

Getting Started

5828-016

ARC® Video Subsystem MPEG 2/4 Decoder Getting Started

ARC® International

European Headquarters
ARC International,
Verulam Point,
Station Way,
St Albans, Herts, AL1 5HE, UK
Tel. +44 (0) 1727 891400
Fax. +44 (0) 1727 891401

North American Headquarters
3590 N. First Street, Suite 200
San Jose, CA 95134 USA
Tel. +1 408.437.3400
Fax +1 408.437.3401

www.arc.com

Confidential and Proprietary Information

Notice

U.S. Government Restricted Rights Legend

Trademark Acknowledgments

5828-016 April 2008

Contents

Chapter 1 — Overview

The ARC® Video Subsystem MPEG 2/4 Decoder for the ARC 700 processor core with the ARC Media Extensions, provides the following codecs:

- MPEG-2 Main Profile and Main Level Decoder, MPEG-2 MP@ML
- MPEG-4 Simple Profile Decoder, MPEG-4 SP
- MPEG-4 Advanced Simple Profile Decoder, MPEG-4 ASP (no GMC)
- H.263 Annex J, I, T, ITU-T Compliant
- MJPEG Decoder

The following sections provide further details on using the codec:

- The Documentation section contains an outline of the online documentation provided with the codec.
- The ARChitect Support section describes how to build the codec from the IP library.
- The MetaWare Toolset Support section describes how to compile and run the codec after it has been built from the ARChitect tool.

Chapter 2 — Documentation

The *ARC Video Subsystem MPEG 2/4 Decoder Bookshelf* is the starting point for all the documentation, and provides links to the following:

- *ARC Video Subsystem MPEG 2/4 Decoder Release Notes*: Containing a summary of the release, and changes since the last release.
- *ARC Video Subsystem MPEG 2/4 Decoder Getting Started*: Containing a summary on using the product.
- *ARC Video Sub-System Codecs Application Programming Interface Reference*: Describing the C Application Program Interface (API) and resource requirements and the basic integration and usage of the codec library.

The following online information is available:

- Online Documentation

Online Documentation

The ARC Video Subsystem MPEG 2/4 Decoder is supplied with built-in on-line documentation that is visible after the IP Library is enabled in the ARChitect list of libraries.

To view the on-line documentation, select **Help > Contents** from the ARChitect menu then select the **IP Library** tab in the help viewer.

For more information on using the help viewer, topic layout and printing topics, select **Help > Help on Help** from the help viewer menu.

Chapter 3 — ARChitect Support

This section describes ARChitect support for the ARC Video Subsystem MPEG 2/4 Decoder. The ARC Video Subsystem MPEG 2/4 Decoder component is built from the IP library using the following ARChitect modes:

- ARChitect Graphical User Interface
- ARChitect Command-Line Option

Once the ARChitect tool has built the ARC Video Subsystem MPEG 2/4 Decoder, the next step is to compile the code using the MetaWare toolset.

ARChitect Graphical User Interface

The ARC Video Subsystem MPEG 2/4 Decoder component is supplied as an ARChitect IP library. To enable the IP library, select **Project > Default Properties > Paths** dialog and add the library file (`.iplib`) to the list of IP Libraries.

After you enable the library, the **MPEG 2/4 Decoder** component appears in the **Video Codecs** tab in the **Components** view of the **Explorer** pane.

To generate the ARC Video Subsystem MPEG 2/4 Decoder software library:

- Add the **MPEG 2/4 Decoder** component to a project that has used the **ARCvideo** template.
- Ensure that the **Build Software** and **Build Test Code** options are enabled in the **Build Sequence** viewer in the **Explorer** pane.
- Perform a project build using the menu option **Project > Build** or click the **Build** button.

The ARC Video Subsystem MPEG 2/4 Decoder software library is built in the following folder:

target_build`/software/video_codecs/MPEG_2-4_Decoder`

When the **VLCHardware** component is selected in the **Topology** view, the following option is available in the **Options** pane:

Legacy (MPEG 2/4) VLC support	To build the codec to make use of the VLC acceleration hardware for MPEG-2/4.

When you select the **MPEG 2/4 Decoder** component in the **Topology** view, the following options are available in the **Options** pane:

ISO GMC support	By default, the codec supports GMC using a fast heuristic which is not fully compliant with the ISO standard. To ensure full ISO compliance, select this option.

Once the ARChitect tool has built the ARC Video Subsystem MPEG 2/4 Decoder, the next step is to compile the code using the MetaWare toolset.

ARChitect Command-Line Option

To enable the ARC Video Subsystem MPEG 2/4 Decoder, use the ARChitect command-line option

-mpeg_decoder

Ensure the **-build_software** and **-build_test_code** command line options are used to correctly generate the codec and associated test code.

The following command line options are also available:

-vlc_has_legacy	To build the codec using the VLC acceleration hardware for MPEG-2/4..
-isogmc	By default, the codec supports GMC using a fast heuristic which is not fully compliant with the ISO standard. To ensure full ISO compliance, select this option.
-libraries	If the ARC Video Subsystem MPEG 2/4 Decoder IP Library is not enabled in ARChitect, you can specify it on the command line with the **-libraries** command-line option.
-template	You can use the **-template** command-line option to specify the **ARCvideo** template.

Once the ARChitect tool has built the ARC Video Subsystem MPEG 2/4 Decoder, the next step is to compile the code using the MetaWare toolset.

Chapter 4 — MetaWare Toolset Support

This section contains the steps for compiling and running a project using the MetaWare toolset.

The MetaWare toolset can compile and run the ARC Video Subsystem MPEG 2/4 Decoder component using the following modes:

- MetaWare Toolset Command Line

The following sections contain further information on configuring and using the codec:

- Compilation Environment
- Configuring the Compilation
- Target-Application Command line

Compilation Environment

The ARC Video Codec uses a makefile to compile the project. Ensure that a suitable **make** utility is installed and in the path. The MetaWare Development Toolkit uses GNU make (**gmake**), which is included in the toolset and selected by default in the MetaWare IDE.

On Windows, ensure that a *remove* command (**rm**) is in the path, to allow **make clean** type commands to run correctly. A remove command, **rm**, is provided in the `mktools/mkbin` folder of the MetaWare Development Toolkit installation folder. Add the *metaware_installation*`/mktools/mkbin` folder to the path if no other *remove* command is available.

Windows users should also ensure that a Perl interpreter is available and installed and has files of extension `.pl` associated with it. Particular Perl scripts are run when dependency files get regenerated, for example after source files have been edited, or after a *make* that removes and regenerates dependency files.

The **make** system uses relative paths from the base installation directory; run it only from the base installation directory.

The **make** system assumes that the MetaWare C/C++ compiler tool chain is installed and is in the path.

Configuring the Compilation

The ARC Video Subsystem MPEG 2/4 Decoder can be compiled to take advantage of hardware acceleration for Huffman decoding operations. You can use ARChitect to rebuild the software library with the support for Huffman decoding operations.

By default, support for Huffman decoding is enabled in the makefile. If the target hardware system does not support VLC hardware acceleration for Huffman, you must edit the makefile.

To disable support for Huffman acceleration, open the file `Makefile` with a suitable text editor and change the following line from:

```
export ENABLE_VLC=true
```

To

```
export ENABLE_VLC=false
```

Target-Application Command line

The supplied example target application is able to open an MPEG 1, 2 or 4 elementary data stream, pass its contents in NAL units to the codec and save the returned decoded YUV frames to a file. The command line interface of the compiled application is as follows:

`mpegdec` *inputfile*`.fmt` *outputfile*`.yuv`

Where *inputfile*`.fmt` is a valid, fully qualified path to an MPEG 2/4 elementary file with a file extension of `m2v`, or `m4v`. The *outputfile*`.yuv` is a valid fully qualified path to the file to be created during decode.

Specify these options to the MetaWare debugger regardless of the target environment (simulator or ARCangel 4 development system).

MetaWare Toolset Command Line

The following section describe how to compile and run projects for the ARC Video Subsystem MPEG 2/4 Decoder using the MetaWare toolset command line:

- Compiling the Codec at the Command Prompt
- Running the Codec at the Command Prompt
- Using the ARCangel 4 Development System at the Command Prompt

Compiling the Codec at the Command Prompt

Ensure the compilation environment is correctly set up before compiling the codec.

To compile the ARC Video Subsystem MPEG 2/4 Decoder, open a command window, go to the folder that contains the makefile, and type the relevant **make** command to invoke your **make** utility, for example:

make

If the local make system does not automatically detect the `Makefile` file in the directory, use the appropriate command line to define the makefile to use, for example:

make –f Makefile

The following make commands can be used to re-compile the library if it has been re-built by the ARChitect tool:

make libclean clean	To clean (i.e. remove object files, executables and libraries).
make libdepclean depclean	To remove dependencies and objects, libraries and executables
make lib	To make just the library.
make all	To make everything.
make libclean clean all	To do a full clean compile (not regenerating dependencies).

make libdepclean depclean all To do a full clean compile (regenerating dependencies).

The MetaWare instruction-set simulator supports the ARC Media Extensions; you can use it to run and debug the example codec application. Alternately, you can run the software on an ARCangel 4 development system using an FPGA XBF image file.

Once the code is compiled the following steps can be carried out:

- Running the Codec at the Command Prompt
- Using the ARCangel 4 Development System at the Command Prompt

Running the Codec at the Command Prompt

To run the simulator successfully, ensure that the ARC Media SIMD Extensions are available. The SIMD extensions are delivered as an optional plug-in to the base MetaWare toolset, and are delivered in the form of a single file `simd.`*OS_specific_file_extension*.

The SIMD extensions file is not included as part of the standard MetaWare installation process. The SIMD extensions file is delivered with the ARC Media Extensions IP library, therefore it is necessary to install the ARC Media Extensions IP library and generate a build using the ARChitect configuration tool.

- If the ARChitect GUI is used, ensure that the **Build Software** and **Build Test Code** options are enabled in the **Build Sequence** viewer in the ARChitect **Explorer** pane to correctly generate the codec and associated test code.
- If the ARChitect command line is used, ensure the **-build_software** and **-build_test_code** command line options are used to correctly generate the codec and associated test code.

The SIMD extensions file is located in the *build_directory*`/iss` folder. For convenience, you can copy the SIMD extension file to the `MetaWare/arc/bin` installation directory.

The following steps outline the process required to execute the codec by using the instruction set simulator.

Run the MetaWare debugger using the following command line options to execute the target file to completion:

> **mdb -simextp=***build_directory***/iss/simd,scm_size=5120 -arc700 -xml=aurora -Xlib -Xmpy –nooptions** *path-to-executable*/**mpegdec** *inputfile***.***fmt outputfile***.yuv**

Where, according to the application command line, *inputfile*`.`*fmt* is a valid, fully qualified path to an MPEG 2/4 elementary file with a file extension of `m2v`, or `m4v`. The *outputfile*`.yuv` is a valid fully qualified path to the file to be created during decode. The codec outputs the specified sequence as YUV 4:2:0.

The following section shows how to use the ARCangel 4 Development System at the command prompt.

Using the ARCangel 4 Development System at the Command Prompt

To run the software on an ARCangel 4 development system:

- Ensure that an ARCangel 4 is connected to the host PC using a parallel cable and that the ARCangel 4 is powered.

- Download (*blast*) the FPGA image file containing the ARC 700 processor with Media Extensions to the ARCangel 4.
- Run the MetaWare debugger using the following command line options, for an example 70MHz build, to execute the target file to completion:

  1 mdb -DLLprop=gclk=70 -hard –jtag –nooptions *path-to-executable*/**mpegdec** *inputfile***.***fmt* *outputfile***.yuv**

- Where, according to the application command line, *inputfile*`.`*fmt* is a valid, fully qualified path to an MPEG 2/4 elementary file with a file extension of `m2v`, or `m4v`. The *outputfile*`.yuv` is a valid fully qualified path to the file to be created during decode. The codec outputs the specified sequence as YUV 4:2:0.