ARC® H.264 Encoder
ARC Video Subsystem Family H.264 Baseline Profile Encoder

Getting Started

5882-007

ARC® H.264 Encoder Getting Started

ARC® International

European Headquarters	North American Headquarters
ARC International,	3590 N. First Street, Suite 200
Verulam Point,	San Jose, CA 95134 USA
Station Way,	Tel. +1 408.437.3400
St Albans, Herts, AL1 5HE, UK	Fax +1 408.437.3401
Tel. +44 (0) 1727 891400	
Fax. +44 (0) 1727 891401	

www.arc.com

5882-007 April 2008

Contents

List of Tables

Chapter 1 — Overview

The ARC® H.264 Encoder is an accelerated encoder for the ARC Video Subsystem Family that complies with Baseline Profile of the ITU H.264 and ISO/IEC AVC standards. The encoder is supplied as an ARChitect IP library.

The following sections provide further details on using the encoder:

- The Installation section outlines the installation requirements.

- The Documentation section contains an outline of the online documentation provided with the encoder.

- The ARChitect Support section describes the use of the subsystem template with the ARChitect tool.

- The MetaWare Toolset Support section describes the use of the encoder with the MetaWare toolset.

Installation

Particular installation instructions for the ARC H.264 Encoder codec and any third party tools are contained in the *README* file that is provided with the installation.

Chapter 2 — Documentation

The *ARC H.264 Encoder Bookshelf* is the starting point for all documentation, and provides links to:

- *ARC H.264 Encoder Release Notes*: Contains a summary of the current release, including changes since the last release.

- *ARC H.264 Encoder Getting Started*: contains a summary on using the product.

- *ARC H.264 Encoder API Reference*: contains details on interfacing to, and controlling, the encoder.

- *ARC Video Subsystem Family Encoders API Programmer's Guide*: provides a basic reference guide to interfacing to this encoder, and the other encoders supported by the ARC Video Subsystem Family.

The following online information is available:

- [Online Documentation](#)

- [Printable Documentation](#)

- [Build-Specific Documentation](#)

Online Documentation

The ARC H.264 Encoder 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.

Printable Documentation

When the ARC H.264 Encoder is built from the ARChitect tool, a set of documentation in the Adobe® portable document format is provided.

The documents for the ARC H.264 Encoder are located in the `docs/pdf` subdirectory of the *target_build*/`software/video_codecs/H264_Encoder` folder.

The PDF documents are also available on the Technical Support site, http://support.ARC.com/.

Build-Specific Documentation

The encoder is deployed by including it into a design using the ARChitect tool. Once ARChitect has performed a build, custom build documentation is available in the target build folder. The following

table summarizes some of the key files generated. ARChitect design flow libraries may also generate additional build files.

Project Summary	`docs/project.html`	This file contains a printable summary of the built project, including register maps, memory maps, instruction lists and configuration information.
ARChitect Build Command	`arc_bld.log`	This is the actual command line that was used to generate the subsystem build
ARChitect Build Configuration File	`build_configuration.txt`	This build configuration file can be re-used at the ARChitect command line with the **-argument_file** command line option to rebuild the project.
ARChitect Build Output Log	`build_output_log.txt`	This is a copy of the messages that were generated in the **Build** view of the ARChitect **Messages** pane.

Chapter 3 — ARChitect Support

This section describes ARChitect support for the ARC H.264 Encoder. The ARC H.264 Encoder codec is built from the IP library using the following ARChitect modes:

- ARChitect Graphical User Interface

- ARChitect Command-Line Option

ARChitect Graphical User Interface

The ARC H.264 Encoder codec 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 **H264 Encoder** component appears in the **Video Codecs** tab in the **Components** view of the **Explorer** pane.

To generate the ARC H.264 Encoder software library:

- Add the **H264 Encoder** component to a project that has used a suitable ARC Video Family template, such as the ARC AV 417V;

- Ensure that the **Build Software** 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 H.264 Encoder software library is built in the following folder:

target_build/`software/video_codecs`

ARChitect Command-Line Option

To enable the ARC H.264 Encoder, use the ARChitect command-line option

 -h264_encoder

Ensure the **-build_software** command line option is used to correctly generate the codec and associated test code.

Chapter 4 — MetaWare Toolset Support

This section describes MetaWare tool support for the ARC H.264 Encoder software. The ARC H.264 Encoder software is supported in the following MetaWare tools:

- MetaWare Compiler Support
- MetaWare Debugger Support

MetaWare Compiler Support

The ARChitect build process will build the encoder as an archive library, and link it into a test harness ELF file suitable for use under the ISS or on an ARCangel 4. This section describes how the encoder may be rebuilt from a command prompt.

The following sections provide further details on compiling the codec:

- Compilation Environment
- Compiling the Encoder and Test Harness

Compilation Environment

The ARC H.264 Encoder 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. The **make** system assumes that the MetaWare C/C++ compiler tool chain is installed and is in the command path search list.

Compiling the Encoder and Test Harness

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

To compile the ARC H.264 Encoder at a Linux command prompt, go to *target_build*/`software/video_codecs` and enter the following command:

make -f Makefile.linux.h264

Likewise, to build the encoder from a Windows command prompt, go to *target_build*/`software/video_codecs` and run the following:

gmake -f Makefile.win32.h264

The build process produces the following two outputs:

- `h264_encoder_app` - the encoder built into a test harness, ready for execution on an ARCangel 4 or under the ISS.
- `lib/libh264_encoder_app.a` - an archive library containing the encoder, ready for linking into end-user applications.

The makefiles support the following targets for use in rebuilding the library and test harness:

`all`	Build the archive library, then the test harness
`lib`	Build the archive library
`clean`	Remove the test harness and its object files
`libclean`	Remove the library object files
`libdepclean`	Remove the library dependency files

After running **make libdepclean**, the next use of the makefile will cause the dependency files to be regenerated.

MetaWare Debugger Support

The output of the compilation process is an executable file that converts YUV data stored in a file to encoded data held in an output file. The executable can be run in two ways:

- Running the Encoder

- Running the Encoder on the ARCangel 4 Development System

An encoder configuration file containing encoding set-up parameters and details of input and output files is also supported. For further information see section Configuration File Format.

Running the Encoder on the Instruction Set Simulator

In its default mode, the Instruction Set Simulator (ISS) can only execute code compiled for a standalone ARC processor. To run the encoder successfully on the ISS, the extension libraries appropriate to your design need to be available. These extensions are delivered as optional plug-ins to the base MetaWare toolset, and are installed into the project by the ARChitect build process. Please refer to the documentation relevant to the media subsystem in question for full details.

To run the encoder test harness on the ISS, run the following command from the *target_build*/`software/video_codecs` directory:

mdb @*target_build*/**iss_config.arg h264_encoder_app.elf -c** *configfile*

Where:

@*target_build*/**iss_config.arg**	The @ command specifies a *file* to be used that contains further command line options, in this case the ISS configuration file containing, amongst other details, the locations of the extension libraries.
configfile	An encoder configuration file containing encoding set-up parameters and details of input and output files. For further information see section Configuration File Format.

Running the Encoder on the ARCangel 4 Development System

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 an ARC Video Family subsystem to the ARCangel 4.

- From the *target_build*/software/video_codecs directory, run the MetaWare debugger using the following command line options, for an example 50MHz build:

  mdb -DLLprop=gclk=50 -hard –jtag -mem=0x20000000 –nooptions h264_encoder_app -c *configfile*

Configuration File Format

The encoder executable reads a configuration file that provides details on how the encoding should be performed. This consists of a series of entries, each on a separate line, each containing a name-value pair, for example:

```
input_yuv   = "C:\YUV\cif\foreman.yuv"
```

A small number of values related to setting up the test harness. These are listed in Test harness settings.

Table 1 Test harness settings

Name	Description
input_yuv	Input file name
output_h264	Destination filename into which encoded stream will be written
output_stats	Destination file into which statistics (if enabled) will be written
output_recon_yuv	Destination file into which reconstructed data (if enabled) will be written.

The remaining values split into two groups: encoding parameters and VUI parameters. The former affect the encoding process, the latter are passed through to the decoding process via the output stream. These are listed in Encoder set-up parameters and VUI parameters, respectively. For a full description of the use and range of acceptable values for each of these, please refer to the *ARC H.264 Encoder Reference*.

Table 2 Encoder set-up parameters

frame_width
frame_height
fixed_quantiser
fixed_quantiser_value
target_bitrate
intra_picture_frequency
code_all_pictures
target_picture_period
input_picture_period
intra_refresh_rate
real_time_rate_control
chroma_qp_index_offset
constrained_intra_pred_flag
number_of_slice_groups_for_p_pictures
disable_deblocking_filter_idc

maximum_packet_size

reaction_multiplier

quality_level

Table 3 VUI parameters

aspect_ratio_info_present_flag

aspect_ratio_idc

sar_width

sar_height

overscan_info_present_flag

overscan_appropriate_flag

video_signal_type_present_flag

video_format

video_full_range_flag

colour_description_present_flag

colour_primaries

transfer_characteristics

matrix_coefficients

chroma_loc_info_present_flag

chroma_sample_loc_type_top_field

chroma_sample_loc_type_bottom_field

timing_info_present_flag

num_units_in_tick

time_scale

Finally, a couple of points to note about the configuration file format:

- The file parsing is case-sensitive. The names of the settings should always be in lower-case, and care should be taken to specify filenames correctly when running on host platforms that have case-sensitive file systems.

- The # symbol indicates that the remainder of the line following it is a comment, and is to be ignored by the configuration file parser.