ucode_interface project

Overview

This repository contains the sourcecode for the *ucode_interface* project. This tool is used to perform the following functions:

i. Connect to a remote target running the `em8xxx_server`.
ii. Load either the **debug** or **release** version of the microcode.
iii. Initialize the microcode interface.
iv. Send the init & play commands.
v. Stream the media file through the bitstream FIFO.
vi. Wait for user to hit 'q' to quit.

Checkout and Build project

This section discusses how to check-out the *ucode_interface* project from the git repository.

Currently this project must be built as a 32bit executable. The project also utilizes the latest features of C++11 thus it requires compiling with a suitable compiler. **gcc-4.9** is available on the Ubuntu 14.04 distribution and is the compiler used to generate the libraries and executables. See the section below regarding any other dependencies of the *ucode_interface* project.

Checking out the project

Two repositories should be checked out. The first is the repository containing the *ucode_interface* sourcecode and the second is the repository containing the pre-built microcode for a variety of platforms.

```
Make a directory to keep the project in:
$> mkdir ucode-interface
$> cd ucode-interface

To check out the source-code repository:
$> git clone ssh://gitmaster@revelation.sdesigns.com/home/gitmaster/gitroot/ucode_interface.git

To check out the microcode repository:
$> git clone ssh://gitmaster@revelation.sdesigns.com/home/gitmaster/gitroot/ucode_repository.git

Make a symbolic link inside the ucode_interface directory to the ucode_repository directory.
$> cd ucode_interface
$> ln -s ../ucode_repository/ ucode
```

Building the project

The *ucode_interface* project employs a novel build system based on the `make` tool and Make scripts. These build scripts are located in the `Build` directory. A text-based GUI based on `dialog` provides a wrapper around the makefiles and it builds all required modules in a variety of manners (static, shared, 32-bit, 64-bit). This build system is also used for many of the tools including the Debugger and the Probe.

The first step is to source the `env.sh` script located in the top-level of the *ucode_interface* project. This script puts various scripts into the path and sets some aliases.

```
$> source env.sh
```

The next step is to build the library components:

```
$> cd Components
$> build.sh
```

This will launch the text-based GUI which will display the status of the build. Once the components are built then we can start to build the interface tool itself.

```
$> cd ../interface4
$> build.sh
```

Now the project is built.

Running the interface tool

`interface4` provides a method of exercising the microcode interface on a remote target, whether an actual chip or a simulation. It communicates with the remote target using the `em8xxx_server` (gbus server) so it can work with any system which provides this TCP/IP server.

The following commandline arguments are supported:

```
--chip / -c chipid          Pass in the chip id. (for instance 8756, 8758, or 8760)
--stream / -s stream        Pass in the media to stream to the interface bitstream FIFO.
--decoder / -d codecid      Pass in the decoder id to specify which codec to use. (for instance mpeg2,
 h264, hevc)
--yuv / -y yuvoutfile       Pass in the name of the output YUV file to create.
--engine / -e engineid      Pass in the engine number to use for the decode.
--mode / -m mode            Pass in 'r' for release microcode, 'd' for debug microcode.
--remote / -r targetspec    Pass in the remote target spec (or use EM8XXX_SERVER by default).
```

Using these options it is possible to start up a decode very easily.

Examples

Here are some examples of commands:

```
./bin/Release/x86/interface4 --chip 8758 --decoder mpeg2 --stream /media/elementary/mpeg2/test100.m2v
--yuv /tmp/test100.yuv -r 10.10.10.12:0 -m d

./bin/Release/x86/interface4 --chip 8758 --decoder mpeg2 --stream /media/elementary/mpeg2/test100.m2v
--yuv /tmp/test100.yuv -r 10.10.10.12:0 -m r

./bin/Release/x86/interface4 --chip 8760 --decoder mpeg2 --stream /media/elementary/mpeg2/test100.m2v
--yuv /tmp/test100.yuv -r 10.10.10.102:0 -m d
```

Supported Targets

- 8756
- 8758
- 8760

Thursday, 03. November 2016 05:33PM