Wiki

Overview

The Common Scripts Library contains a parametrizeable, extensible, versatile collection of scripts to use with the Altera Nios Command Shell. These scripts can easily be integrated into projects in a matter of minutes, providing a powerful toolset.

Basic Integration

Integration of the common scripts library is done as follows:

1. The appropriate release of the Common Scripts Library **must** be located in the tsbs/common_scripts_library using the standardized hierarchical design method
2. In the tsb/ip/scripts directory of the encompassing module, the file "project_settings.ini" needs to be defined (see [[The Project Parameter File]])
3. In the tsb/ip/scripts directory, a script must be present for each script used in the Common Scripts Library, with the following contents:

```
#!/bin/bash
script_name=$(basename $BASH_SOURCE)
source ../../../tsbs/common_scripts_library/tsb/ip/scripts/${script_name} "$@"
```

> *For example, if usage of the script do_altera_hardware_load.cmd is desired, a file with the name do_altera_hardware_load.cmd needs to be present in the tsb/ip/scripts directory with the above contents. Note that the content is script-independent.*

Running Scripts

In general, running scripts is done via the "source" method, i.e.:

```
source do_altera_hardware_load.cmd
```

Parameters can also be passed to many scripts, e.g.

```
source do_altera_hardware_load.cmd  -g
```

The project_settings.ini file

The project_settings.ini file is an easily editable INI file that is parsed by the common scripts library. In contains project-dependent settings that are used by the various scripts. See:
[[The Project Parameter File]]

List of scripts

For a list of scripts with usage details, see the following:

Custom scripts

It is easy to build upon the common scripts library in order to easily construct project-specific custom scripts. These custom scripts should also reside in the tsb/ip.scripts directory. For example, in edevel00396, the script do_burn_firmware.cmd is:

```
source do_program_maxv_flash_w_writer.cmd
source do_program_cfi_flash.cmd
source do_program_maxv_flash_w_loader.cmd
```

Where do_program_cfi_flash.cmd is:

```
#!/bin/bash
source ../../../tsbs/common_scripts_library/tsb/ip/scripts/do_program_main_fpga_flash.cmd  "$@"
```

and do_program_maxv_flash_w_writer.cmd is:

```
#!/bin/bash
source ../../../tsbs/common_scripts_library/tsb/ip/scripts/do_program_maxv_flash_w_alternate_file.cmd "$@"
```

and do_program_maxv_flash_w_loader.cmd is:

```
#!/bin/bash
source ../../../tsbs/common_scripts_library/tsb/ip/scripts/do_program_maxv_flash.cmd "$@"
```

Where the appropriate variables used by the common library scripts are, as usual, defined in the project_settings.ini file

Examples

Examples of the usage of the scripts library can be found in:

edevel00365 - DEAP firmware on 3xFMC Stratix IV board - complicated example of script usage
edevel00396 - 32 Channel ADC on Arria V development board - complicated example of script usage
edevel00243 - MaxV firmware for 3xFMC board - simple example of script usage

[[List of Scripts]]

List of Scripts

User Scripts

Here is a brief description of some useful scripts. Further details can be easily obtained by opening the scripts in a text editor.

convert_pof_to_rbf.cmd - convert a POF file to an RBF file (base filename given as parameter)
convert_sof_to_rbf.cmd - convert a SOF file to an RBF file (base filename given as parameter)
do_altera_hardware_load.cmd - Load the hardware SOF file to the FPGA
do_app_nios_load.cmd - load the ELF file to the FPGA, for Cygwin
do_app_nios_load_linux.cmd - load the Nios ELF file to the FPGA, for Linux
do_app_nios_load_no_reset.cmd - load the Nios ELF file to the FPGA, for Linux, without resetting the processor
do_bootloader_nios_load.cmd - load the Nios bootloader ELF file
do_compile_altera_rtl.cmd - Compile the Altera Quartus project (currently the same as do_compile_altera_rtl_64bit.cmd)
do_compile_altera_rtl_64bit.cmd - Compile the Altera Quartus project using Quartus 64-bit tools
do_connect_to_jtag_uart.cmd - Connect to the JTAG UART of the main Nios processor, for Cygwin
do_connect_to_jtag_uart_linux.cmd - Connect to the JTAG UART of the main Nios processor, for Linux
do_connect_to_jtag_uart1.cmd - Connect to the JTAG UART of the bootloader Nios processor, for Cygwin
do_connect_to_jtag_uart1_linux.cmd - Connect to the JTAG UART of the bootloader Nios processor, for Linux
do_convert_elf_to_rbf.cmd - convert ELF file to RBF file
do_convert_original_sof_to_rbf.cmd - converts the project SOF file, as defined by the project settings file
do_convert_pof_to_rbf.cmd - identical to convert_pof_to_rbf.cmd
do_get_project_settings.cmd - parses the project_settings.ini file; used by the various scripts and can be used to help write new scripts
do_open_main_qsys.cmd - opens the main QSYS file, as defined in the project settings file, with the QSYS heap size defined by the project settings file
do_open_qsys.cmd - opens a QSYS in the tsb/ip/rtl directory, but does not load a QSYS file
do_open_quartus.cmd - opens Quartus with the project as defined in the project settings file
do_program_aux_fpga_flash.cmd - programs the flash for the auxiliary FPGA devices (e.g. Spartans on the 3xFMC card), parameters defined in the project_settings.ini file
do_program_main_fpga_flash.cmd - programs the flash for the main FPGA device (e.g. Stratix IV on the 3xFMC card), parameters defined in the project_settings.ini file
do_program_maxv_flash.cmd - programs the flash for the MAXV device, parameters defined in the project_settings.ini file
do_program_maxv_flash_w_alternate_file.cmd - programs the flash for the MAXV device, using an alternate file as defined in the project_settings.ini file
do_run_custom_system_console.cmd - runs Altera's system console tool with the custom TCL script custom_system_console_script.tcl that is present in the encompassing file's tsb/ip/scripts directory
do_run_simple_system_console.cmd - runs Altera's system console tool, directory set to the tsb/ip/rtl directory
do_run_system_console.cmd - runs the DASHING system-console based tool
envchk.sh - run this as "./envchk.sh" (without sourcing): prints out the path to Quartus

Advanced Scripts

Any scripts not listed as user scripts should be considered advanced scripts. See [[Advanced Scripts]] for more information.

The Project Parameter File

The project_settings.ini file is an easily editable INI file that is parsed by the common scripts library. In contains project-dependent settings that are used by the various scripts. In must be located in the tsb/ip/scripts directory of the encompassing module.

Example project_settings.ini File

An example of the project_settings.ini file is show here for DEAP (edevel00365)

```
[project]
main_fpga_project_filename_base=3xFMC_Generic
main_fpga_project_programming_file_extension=sof
main_fpga_project_elf_filename_base=allserver_sfp
bootloader_filename_base=deap_boot_loader
bootloader_elf_base=boot_loader
daq_nios_filename_base=Application_new
main_fpga_device_index=1
main_fpga_memory_space_start_addr=0
default_usb_cable_index=0
application_nios_instance_index=0
application_nios_associated_jtag_uart_instance_index=0
bootloader_nios_instance_index=1
bootloader_nios_associated_jtag_uart_instance_index=1
quartus_project_location_directory=../../../tsb/ip/rtl
quartus_project_postcompile_sof_location_directory=../../../tsb/ip/rtl/output_files
quartus_project_nios_app_software_directory=../../../tsb/ip/rtl/software
quartus_project_daq_nios_app_software_directory=../../../tsbs/deap_vme_and_daq/tsbs/trigger_and_daq/tsb/ip/rtl/software/
cof_filename_for_programming_main_fpga_flash=configuration_file_for_programming_stratix_cfi_flash_with_sof_and_elf
cdf_filename_for_programming_main_fpga_flash=program_stratix_cfi_flash
rbf_filename_for_programming_main_fpga_flash=stratix_sof_and_elf_image
cropped_rbf_filename_for_programming_main_fpga_flash=stratix_sof_and_elf_image_cropped
map_filename_for_programming_main_fpga_flash=stratix_sof_and_elf_image
map_row_header_for_max_address_in_main_fpga_flash=Page_0
cdf_filename_for_programming_maxv_flash=program_maxv_flash_normal
cdf_alternate_filename_for_programming_maxv_flash=program_maxv_flash_fallback
cof_filename_for_programming_aux_fpga_flash=configuration_file_for_programming_flash_for_three_spartans_deap_fmc2_adc_fmc1_cg
c_fmc1_cgen_fmc0_trig
cdf_filename_for_programming_aux_fpga_flash=program_spartan_hex_files
pof_filename_for_programming_aux_fpga_flash=spartan_images_deap_fmc2_adc_fmc1_cgen_fmc0_trig

[qsys]
main_fpga_flash_base_addr=0x54000000
main_fpga_flash_end_addr=0x58000000
java_heap_size_for_qsys=2048M
main_qsys_filename=NIOS_ApplicationProcessor.qsys
```

DEAP is a rather complicated example with multiple FPGAs. A simpler example is for the MaxV project for DEAP (edevel00243). The project_settings.ini file for that project is:

```
[project]
main_fpga_project_filename_base=maxv_3xfmc
main_fpga_project_programming_file_extension=pof
main_fpga_device_index=2
main_fpga_memory_space_start_addr=0
default_usb_cable_index=0
quartus_project_location_directory=../../../tsb/ip/rtl
quartus_project_postcompile_sof_location_directory=../../../tsb/ip/rtl/output_files
cdf_filename_for_programming_maxv_flash=program_maxv_flash

[qsys]
java_heap_size_for_qsys=1024M
main_qsys_filename=maxv_qsys_spi_bridge.qsys
```

Usage of the project_settings.ini file in the Common Scripts Library

The usage of the project_settings.ini file in each script is done as follows. For example, let's take a look at the script do_altera_hardware_load.cmd, which loads the SOF file into the FPGA.

```
source do_get_project_settings.cmd
nios2-configure-sof --device $main_fpga_device_index "$@" ../exe/$main_fpga_project_filename_base\.sof
```

The usage of the project_settings.ini file is done in two steps:

1. The command:

```
source do_get_project_settings.cmd
```

calls the special script "do_get_project_settings.cmd" that parses in "project_settings.ini" file and puts all of the settings in BASH variables.

2. The script uses the various variable in the commands. For example the line:

```
nios2-configure-sof --device $main_fpga_device_index "$@" ../exe/$main_fpga_project_filename_base\.sof
```

Uses the variable "main_fpga_device_index" to define the device in the JTAG chain where the FPGA resides, and "main_fpga_project_filename_base" as the filename of the "sof" file.

Note that "$@" means the parameters passed to the script, so additional parameters can be passed via that construct.