ALPHA DATA

ADM-XRC Gen 3 SDK 1.2.0 for Linux Release Note

Introduction

This release note accompanies the ADM-XRC Gen 3 for Linux. The latest version of this SDK can be found at:

ftp://ftp.alpha-data.com/pub/admxrcg3/linux

For support, send e-mail to support@alpha-data.com

Operating systems supported

This release of the ADM-XRC Gen 3 SDK supports the following operating systems:

- Linux
- GNU/Linux distribution with 2.6.x kernel

Hardware supported

This release of the ADM-XRC Gen 3 SDK supports the following Alpha Data hardware:

- ADM-XRC-6TL
- ADM-XRC-6T1

Related products

ADB3 Driver for Linux is required to run the examples in this SDK. We recommend using the latest version of the driver that is available from:

ftp://ftp.alpha-data.com/pub/admxrcg3/linux

License Agreement

The license agreement for this software is available in the folder where the software is unpacked, in the file **license.txt**. Please contact Alpha Data if alternative licensing conditions are required.

Alpha Data reserves the right to use a different license agreement for future releases of this software.

Installation instructions

This release of the SDK is distributed in source code form as a tarball (.tar.gz file extention). Please refer to the README file inside the tarball for instructions on how to configure and build the examples within.

Known issues

Incomplete documentation

Documentation of the SDK is incomplete: example applications are not fully documented; documentation for common HDL components is missing.

ISE Project Navigator projects not included

Project Navigator projects are not included in this release, but in order to accomodate users who are accustomed to using the Project Navigator GUI for FPGA development, they will be added in a future release.

Example code for on-board memory

Example C code, HDL code and documentation for using the on-board memory banks on third generation Alpha Data hardware is not included. Xilinx™ Memory Interface Generator (MIG) can be used; contact support@alpha-data.com for details.

Hardware monitoring may require 'pumping'

In PCI-E firmware revisions 0x00 - 0x01 of the ADM-XRC-6TL and ADM-XRC-6T1, the hardware monitoring logic does not automatically trigger conversion cycles in the LM87 system monitor chip. As a result, the values displayed by the 'monitor' and 'sysmon' utilities will always be zero. As a workaround, a script can be run in the background to 'pump' the LM87 interface at intervals of one second.

Linux bash script

```
#!/bin/bash
for-((;;))~;-do
    echo~0x42D0000~|~$ADMXRC3_SDK/apps/linux/dump/dump~$*~wd~2~0x400~4
    sleep~1
done
```

Windows batch file:

```
@ECHO~ON
:loop
ECHO~0x42D0000~|~*%ADMXRC3_SDK%\bin\win32\x86\dump.exe~*%*~wd~2~0x400~4
CHOICE~/N~/D~Y~/C~YN~/T~1~>NUL
GOTO~loop
```

These scripts periodically write the value 0x42D0000 to offset 0x400 in window 2 (Model-specific registers), which commands an LM87 conversion cycle to be performed. Changing the interval to be less than one second is not recommended as the LM87 requires approximately 0.5s to complete a conversion cycle.

PCI-E firmware revision 0x02 and later of the ADM-XRC-6TL and ADM-XRC-6T1 do not require this workaround.

Release history

Release 1.2.0

The Xilinx™ ISE 12.2 toolset was used to generate the prebuilt bitstreams (.BIT) files in this release of the SDK.

Corrections:

1. General:

    (a) Corrected an issue where file permissions in previous releases of the package were not appropriate for the type of each file. Changed the uid and gid of the owner of the files in the package to 0 (i.e. root).

2. Example applications:

    (a) Fixed a bug in the FLASH utility where endian conversion was omitted for the 32-bit boot flag, resulting in the boot flag being set incorrectly on big-endian machines.

    (b) Fixed a memory leak in sampleParseCommandLine when it gets the ADMXRC3_SDK environment variable.

3. Example HDL designs:

(a) Modelsim .do scripts now work correctly when executed in a Linux environment, as the 'gen_today_pkg.bat' and 'gen_today_pkg.bash' scripts have been replaced by a TCL script 'gen_today_pkg.tcl'.

(b) When executing 'make install' for the example FPGA designs in order to copy generated .BIT files into the $ADMXRC3_SDK/bit/ directory, spaces in the value of $ADMXRC3_SDK are now tolerated.

Enhancements:

4. General:

(a) Added support for VxWorks hosted on Windows or Linux.

5. Documentation:

(a) The SDK User Guide now includes information about VxWorks support and the VxWorks example applications.

6. Example applications:

(a) Added VxWorks versions of most example applications.

(b) Added calls to sampleCleanupCommandLine to the example applications so that they no longer leak memory. This doesn't matter much unless using VxWorks, since VxWorks does not automatically reclaim leaked memory when certain types of program terminate.

(c) Added command-line options to the INFO utility for showing information about I/O modules, Flash memory banks and sensors.

(d) Changed the SIMPLE example to exit on CTRL-Z (Windows) or CTRL-D (Linux & VxWorks) instead of entering 0x55AA.

7. ADMXRC3 API header files:

(a) The header file <admxrc3.h> can now be included in a VxWorks application.

8. Example HDL designs:

(a) The way that the MPTL clock and reference clock are input at the top level of the SIMPLE FPGA design has been simplified. The record type containing a bundle of possible clocks, of which only one clock is used, has been replaced by a single clock input. The .UCF file ensures that the correct device pins are used for this clock.

(b) The clock generation block blk_clocks.vhd in the UBER FPGA design has been improved:

   • The MMCM parameters have been changed to allow a greater range of frequencies to be output.

   • Two outputs intended for off-chip memory clocking have been added, one being 2x the frequency of the other.

   • The pll_usr_clk signal has been changed to 80 MHz and renamed pll_reg_clk to better reflect its usage, since it is used for clocking the register blocks in the design.

   • The way the clock for the MPTL core is input from I/O pins has been simplified.

(c) The autogenerated VHDL file containing the build date for the UBER FPGA design is now generated in the directory specific to the model being targetted, and its filename contains the model-device combination. This should (in theory) permit simultaneous builds of an FPGA design for more than one model-device combination without risk of the builds interfering with each other. For example, for UBER targetting an ADM-XRC-6T1 with a 6VLX365T device, the generated file is $ADMXRC3_SDK/hdl/vhdl/examples/uber/admxrc6t1/today_pkg_admxrc6t1_6vlx365t.vhd.

(d) The MPTL cores for the ADM-XRC-6T1 in $ADMXRC3_SDK/hdl/vhdl/common/mptl/admxrc6t1 have been updated. FPGA designs using these cores should be rebuilt.

    (e)   Some model-specific signals used at the top level of the example FPGA designs have been combined into the new record types mptl_sb_b2t_t and mptl_sb_t2b_t, representing sideband signals related to the MPTL. This change abstracts these bundles of signals as the details are normally not of interest to end users.

    (f)   The MPTL wrapper mptl_if_target_wrap has a new input signal ocp_ready. This signal is asserted by the target FPGA design to indicate that it is ready to accept OCP transactions. The purpose of the signal is to hold off software on the host, by delaying return from the ADMXRC3_ConfigureFrom* functions, until the target FPGA is ready (i.e. PLLs locked, IODELAYs callibrated etc.).

Release 1.1.0

This is the first release of the ADM-XRC Gen 3 SDK for Linux.