ALPHA DATA

ADM-XRC Gen 3 SDK 1.3.1 for Linux Release Note

Introduction

This release note accompanies the ADM-XRC Gen 3 SDK 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 not complete.

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.

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 ((;1;)) ; 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

NOTE: in the notes below, $ADMXRC3_SDK and %ADMXRC3_SDK% are equivalent, meaning the path that is the root of the SDK after installation; for example /opt/admxrcg3_sdk-1.3.1.

Release 1.3.1

Note: The Xilinx ISE 12.4 toolset was used to generate the prebuilt bitstreams (.bit files) in this release of the SDK.
This release supersedes beta release 1.3.0b1.

New behavior:

1. Bi-architecture build of the example applications is no longer performed by default, as most 64-bit Linux distributions do not install (by default) the necessary compatibility packages for building 32-bit binaries. To build both 32-bit and 64-bit binaries for the API libraries, specify '-biarch yes' when running the 'configure' script.

Corrections:

2. Example applications:
   (a) Fixed a bug in remaining() function used in DUMP and VPD utilities; caused assertion traps in Windows debug builds.

   (b) Fixed the behavior of the "Device" Combo Box in the Windows version of the SYSMON utility so that when devices are added, removed, disabled or enabled (e.g. using Device Manager), it correctly reflects the changes.
   (c) Command line application framework in $ADMXRC3_SDK/apps/src/common/:
      - Fixed a problem where applications built for Linux that expect UNIX pathnames can have the leading / in a path interpreted as the start of an option rather than as a path.

3. Example HDL designs:
   (a) Fixed a bug in the adb3_ocp_ocp2ddr3 component present in the SDK beta release (1.3.0b1). Please use the updated version of this component (0.3, 05 May 2011) included in this version of the SDK.

Enhancements:

4. General:
   (a) Added support for DDR3 memory on models ADM-XRC-6TL and ADM-XRC-6T1.
5. Documentation:
   (a) The User Guide now contains outlines of the components that are the common HDL modules in $ADMXRC3_SDK/hdl/vhdl/common/. Full details of the operation and usage of these components is scheduled for the next release of the SDK.
6. Example applications:
   (a) The FLASH utility now has the "info" command, which displays information about the Flash memory and then exits. Also added comments to source code that detail the Flash address map on supported models.
   (b) The DUMP utility now displays addresses within the specified window rather than the application's buffer virtual addresses.
   (c) Added a host-driven memory test application MEMTESTH, which performs a test of the memory banks on supported models.
   (d) The MONITOR utility now accepts an option which specifies how many times to perform measurements, so that (for example) a single set of measurements can be displayed.
   (e) The SIMPLE example now exits when CTRL-D (Linux) is pressed or CTRL-Z is entered (Windows), instead of requiring 0x55AA to be entered.
   (f) Command line application framework in $ADMXRC3_SDK/apps/src/common/:
      - The FPGA configuration code from the command-line parsing framework code in $ADMXRC3_SDK/apps/src/common/args.c has been moved into a separate file $ADMXRC3_SDK/apps/src/common/fpga.c, so that args.c itself has no dependency on the ADMXRC3 API.
   (g) Various changes within the apps/src/ tree improve portability of source code, to facilitate building the applications for Windows using non-Microsoft compilers such as **gcc**.
7. ADMXRC3 API header files:
   (a) Added the ADMXRC3_*DMA*Ex functions, which enable a 64-bit local address (i.e. OCP address) to be passed.
   (b) Improved portability for non-Microsoft compilers such as **gcc**.
8. ADMXRC3 API import libraries (Win32):
   (a) Updated to include the functions ADMXRC3_*DMA*Ex and ADMXRC3_*DMA*Ex, which enable a 64-bit local address (i.e. OCP address) to be passed.
   (b) Added GCC-compatible libraries in $ADMXRC3_SDK/lib/win32gnu.

9. Example HDL designs:
   (a) 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.

   The MPTL cores for the ADM-XRC-6TL in **$ADMXRC3_SDK/hdl/vhdl/common/mptl/admxrc6tl have been updated**. FPGA designs using these cores should be rebuilt.

   (b) Added support for full MPTL simulation. Previously, only OCP-only simulation was available.

   (c) Added a new folder **$ADMXRC3_SDK/hdl/constraints/**, which contains required and recommended synthesis constraints (.xcf) and implementation constraints (.ucf) for FPGA designs. Currently it contains mandatory constraints for the MPTL core and the Xilinx MIG v3.6 DDR3 SDRAM interface.

   (d) Added a new package **adb3_ocp_comp**, containing general purpose component definitions, to **$ADMXRC3_SDK/hdl/vhdl/common/adb3_ocp/**.Added new components **adb3_ocp_mux_nb**, **adb3_ocp_ocp2ddr3_nb**, **adb3_ocp_retime_nb**, and **adb3_ocp_split_nb** to **adb3_ocp_comp** package.

   (e) Added a new folder **$ADMXRC3_SDK/hdl/vhdl/common/mem_apps/**, which contains the on-board memory application library.Added new component **blk_mem_test** to the **mem_apps** library.

   (f) Added a new folder **$ADMXRC3_SDK/hdl/vhdl/common/mem_if/**, which contains on-board memory interface code.

   Added new package **mem_if_pkg**.

   Added a new folder **$ADMXRC3_SDK/hdl/vhdl/common/mem_if/ddr3_sdram/**, which contains code for DDR3 SDRAM on-board memory interfaces.

   Added a new folder **$ADMXRC3_SDK/hdl/vhdl/common/mem_if/ddr3_sdram/mig_v3_6/**, which contains the generated Xilinx MIG v3.6 DDR3 interface files and scripts to generate those files.

   Added new component **ddr3_if_bank_v3_6**, which is a wrapper for the Xilinx MIG v3.6 DDR3 interface.

   (g) Added a new folder **$ADMXRC3_SDK/hdl/vhdl/common/mem_tb/**, which contains on-board memory model library.

   Added a new folder **$ADMXRC3_SDK/hdl/vhdl/common/mem_tb/ddr3_sdram/**, which contains DDR3 SDRAM model code.

   Added new component **ddr3_sdram** to the mem_tb library.

   (h) **Uber** example design:
     - Added new blocks **blk_bram**, **blk_clock_freq**, **blk_dma_switch**, **blk_ds_mem_reg**, **blk_mem_app**, **blk_mem_if**, **test_uber_dma_1ch_nb** to **$ADMXRC3_SDK/hdl/vhdl/examples/uber/common**.
     - Added connection of DMA OCP channel 0 to all banks of DDR3 SDRAM using the OCP switching block **blk_dma_switch** and the DDR3 SDRAM bank interface block **ddr3_if_bank_v3_6**. Added connection of all DMA channels to **blk_bram** using the OCP switching block **blk_dma_switch**. All banks of on-board memory and the **blk_bram** share the common DMA OCP channel 0 address space.
     - Added connection of the Direct Slave OCP channel to all banks of DDR3 SDRAM. Direct Slave access to the DDR3 SDRAM banks is via a 2 MiB window augmented by the **BANK** and **PAGE** registers in block **blk_ds_mem_reg**.
     - Added DDR3 SDRAM signals to the design's top level.
     - Combined top-level GPIO signals into a single record.
     - MPTL timing constraints have been moved from .ucf files to .xcf files.

- Added DDR3 SDRAM timing and pin LOC constraints to .ucf files.
- Added **CONFIG STEPPING = 0** constraints to .ucf files, so that generated .bit files should be compatible with any device stepping.
- Changed the names of the clock signals output by **blk_clocks** to better reflect their purpose, and added a new clock used by the DDR3 SDRAM interfaces.
- The DMA test in the testbench has been changed from 2-channel blocking to 1-channel non-blocking.
- The testbench now instantiates several instances of a DDR3 SDRAM device model, and performs tests of the DDR3 SDRAM interfaces.

(i) **Simple** example design:

- Added **CONFIG STEPPING = 0** constraints to .ucf files, so that generated .bit files should be compatible with any device stepping.

Known release-specific issues:

10. Example HDL designs:

    (a) The **Uber** example design sometimes fails to meet timing constraints when placed and routed using the Xilinx ISE tools. The workaround is to rebuild the design, which usually results in timing closure. In Windows, change directory to **%ADMXRC3_SDK%\hdl\vhdl\examples\uber** and execute "nmake clean_*<model>*_*<device>* bit_*<model>*_*<device>*". In Linux, change directory to **$ADMXRC3_SDK/hdl/vhdl/examples/uber** and execute "make clean_*<model>*_*<device>* bit_*<model>*_*<device>*".

    This issue is scheduled to be corrected in the next release of the SDK.

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.

Page Intentionally left blank.