ADB3 Driver 1.4.10 for Windows Release Note

Introduction

This release note accompanies the ADB3 Driver for Windows. The latest version of this driver can be found at:

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

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

Operating systems supported

This release of the ADB3 Driver supports the following operating systems:

- Microsoft Windows 2000
- Microsoft Windows XP, 32-bit and 64-bit editions
- Microsoft Windows Server 2003, 32-bit and 64-bit editions
- Microsoft Windows Vista, 32-bit and 64-bit editions
- Microsoft Windows 7, 32-bit and 64-bit editions
- Microsoft Windows 8, 32-bit and 64-bit editions

Hardware supported

This release of the ADB3 Driver supports the following Alpha Data hardware:

- ADM-XRC-6TL
- ADM-XRC-6T1
- ADM-XRC-6T-DA1
- ADM-XRC-6TGE
- ADM-XRC-6T-ADV8
- ADPE-XRC-6T and ADPE-XRC-6T-L
- ADPE-XRC-6T-ADV
- ADM-XRC-7K1
- ADM-XRC-7V1

License Agreement

Please refer to the file **license.rtf** within this software package for licensing terms. 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 driver is distributed in binary form as a Windows ZIP (.zip file extention). To install it, log on as a user with Adminstrator privileges and follow these steps:

1. Unzip the ZIP file to a convenient location on a local hard disk (this is recommended to avoid User Account Control issues on Vista and later).
2. Open the folder into which the ZIP file was extracted, which should contain a file **adb3.inf** (among others). Then, run the appropriate installer:
   - If the operating system is a 32-bit edition of Windows, launch **install.exe**.
   - If the operating system is a 64-bit edition of Windows, launch **install64.exe**.
3. A dialog box should appear, showing progress. When the 'Finish' button appears, click it to finish installation.
4. Optionally, open Windows Device Manager and verify that the expected number of Alpha Data Reconfigurable Computing devices are present, and that none of them are in an error state (yellow exclamation mark over the corresponding icon).

In Windows Vista and later, User Account Control prompts may appear during the above steps. These prompts should be confirmed.

VPD write-protection mechanism

The VPD write-protection mechanism described in the ADM-XRC Gen 3 SDK User Guide is implemented as of release 1.1.0. To enable write-to-VPD, the following DWORD registry value must be nonzero:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\abd3\Parameters\EnableVpdWrite

This value is checked each time a call to ADMXRC3_WriteVPD is made with valid parameters, so changes to this registry value take effect immediately. If the registry value does not exist, the driver considers it to be zero (write-to-VPD disabled).

Security considerations

By default, only a user with administrative privileges can open a device using the ADMXRC3 API and use the full functionality of the API. This is intentional, as the bus interface in third generation Alpha Data hardware is capable of reading and writing arbitrary addresses. As of release 1.1.0, the driver applies the following SDDL string to devices via the driver's .INF file **adb3.inf**:

```
HKR,,Security,,"D:P(A;;GA;;;SY)(A;;GA;;;BA)(A;;GR;;;BU)"
```

An explanation of Windows SDDL strings is outside the scope of this document, but the above string means that the system and administrators have full control, while normal users have read access. When a user-mode process attempts to open a device, the driver verifies that either (a) the user-mode process is opening the device in "passive" mode (i.e. read-only), or (b) the user owning the process is an administrator. If neither are true, the driver rejects the attempt to open the device. In the case of (a), the driver restricts the user-mode process to using a subset of the device's functionality so that the device cannot be used to compromise system security.

Common buffer support

Beginning with release 1.4.4, the driver can create one or more "common buffers" at startup. The main purpose of this feature is supporting applications that use Direct Master data transfer, such as an ethernet-style I/O interface where the sizes and arrival times of packets of data are not known in advance by software running on

the host. These buffers have the following characteristics:

- Persist until the driver is stopped.
- Guaranteed aligned to a specified power-of-2 address boundary.
- Allocated from the appropriate pool of memory in order to be contiguous and visible to bus-master devices.
- Can be mapped into the virtual address space of a user-mode process; see "ADMXRC3 API Specification 1.5.0" or later for details of the new ADMXRC3 API functions relating to common buffers.

The driver parameter **PrimaryCommonBufferCount** determines the number of common buffers allocated; the default is zero, meaning that no common buffers are allocated by default. The parameter **PrimaryCommonBufferSizeLow** determines the size in bytes of each common buffer; the default is 64 kiB (0x10000). The parameter **PrimaryCommonBufferAlignment** determines the address boundary size to which each common buffer is guaranteed to be aligned; the default is 16 bytes (0x10).

Known issues

Downgrading to an earlier version

Due to Windows driver versioning rules, if the running driver is downgraded to an earlier version and new hardware is plugged in, Windows will prefer to reinstall thehighest-numbered signed driver in the system, and may even do so without presenting any prompts to users. When this occurs, it is necessary to install the downgraded driver again using the installer program from the driver package (**install.exe** or **install64.exe**).

A permanent solution available in Windows Vista and later is to uninstall the later driver along with devices via the Device Manager, and then install the earlier driver using the installer program.

In versions of Windows before Vista, Windows generally prompts the user when new hardware is detected (unless a WHQL-signed driver exists), so the choice exists at that point to select the downgraded driver instead of accepting the driver preferred by Windows.

Fixed-local addressing DMA transfers

The flag ADMXRC3_DMA_FIXEDLOCAL when used with the DMA functions in the ADMXRC3 currently has no effect for Gen 3 hardware.

Release history

Release 1.4.10

This release implements ADMXRC3 API Specification version 1.6.0.

Enhancements:

1. Added new ADMXRC3 API functions: ADMXRC3_GetDeviceStatus and ADMXRC3_ClearDeviceErrors.
2. Added support for ADM-XRC-7K1 revision 2 board.
3. Added support for ADM-XRC-7V1 revision 2 board.
4. Added support for devices implemented using generic ADB3 core in target FPGA (Vendor ID = 0x4144, Device ID = 0xADB3, Subsystem Vendor ID = 0x4144, Subsystem Device ID = 0x0000).
5. When a generic ADB3 device (such as an ADB3 core in a target FPGA) is found, now counts number of DMA engines by examining registers in BAR0, instead of assuming 0. This means that DMA engines are

exposed by the ADMXRC3 API when available.

Corrections:

6 Corrected sensor names for ADM-XRC-7V1.

Release 1.4.9

This release implements ADMXRC3 API Specification version 1.5.1.

Enhancements:

1 Added support for the ADM-XRC-6T-DA1 and ADM-XRC-7K1.

2 Added preliminary support for the ADM-XRC-7V1.

Corrections:

3 The implementation of ADMXRC3_MapWindow and ADMXRC3_UnmapWindow has been revised, in order to behave sensibly for the case where some parent process creates a mapping, then spawns a child process that inherits a device handle, and then terminates before the child process terminates.

4 On the ADPE-XRC-6T(-L), the debug message about an AVR timeout (visible when kernel debugger is running) has been eliminated. The driver now correctly gets the AVR uC firmware version from the AVR uC instead of timing out.

5 On the ADPE-XRC-6T(-L), if a board has a value of 0 (which is always invalid) in VPD for the SI5338 reference clock frequency, works around it by changing it in the in-memory copy of the data to 25000000.

6 On the ADM-XRC-6T-ADV, the second bank of Flash memory (dedicated to target FPGA 1) can now be accessed successfully.

7 Fixed an unkillable thread hang that can occur when calling ADMXRC3_ReadVPD with a VPD addresses of 0x100 or above on the ADPE-XRC-6T(-L), ADM-XRC-6T-ADV8 and ADPE-XRC-6T-ADV.

Release 1.4.6

This release implements ADMXRC3 API Specification version 1.5.0.

Enhancements:

1 Added preliminary support for ADPE-XRC-6T-ADV.

Corrections:

2 Added support for FMCs fitted to ADPE-XRC-6T(-L). Now reports FMC information via ADMXRC3_GetModuleInfo for ADPE-XRC-6T(-L) as expected when an FMC is fitted.

3 ADMXRC3_ReadSensor now correctly reports the values of sensors whose values are negative (e.g. a temperature sensor whose reading is less than 0 deg. C) instead of a large positive value.

4 Corrected name of sensor 1 on ADM-XRC-6T-ADV8; was "12V supply rail", now "5V/12V XMC VPWR rail"; setting driver parameter "Admxrc6tadv8CompatSensor1Name" to 1 overrides this and causes the old name to be returned by the driver.

5 On the ADPE-XRC-6T(-L), fixed clock generator 1 being incorrectly mapped to SI5338 multisynth 0; now mapped to multisynth 1.

Release 1.4.5

This release implements ADMXRC3 API Specification version 1.5.0.

Corrections:

6 Fixed a crash that can occur when ADMXRC3_Unlock is called with an invalid value for the ADMXRC3_BUFFER_HANDLE parameter.

7 Fixed a crash when multiple queued DMA transfers are cancelled, by ADMXRC3_Cancel or by killing threads, within a small window of vulnerability around to the point at which the DMA transfer would complete normally were it not cancelled.

Release 1.4.4

This release implements ADMXRC3 API Specification version 1.5.0.

Enhancements:

1 Added common buffer functionality with the following ADMXRC3 API functions:

- ADMXRC3_GetCommonBuffer
- ADMXRC3_GetCommonBufferCount
- ADMXRC3_MapCommonBuffer
- ADMXRC3_UnmapCommonBuffer

Corrections:

2 Fixed a race condition that could cause a crash every few hours of constant large DMA transfers (of size 64 physical pages or more) in a typical SMP machine.

3 Corrected the scaling factors for sensors 1 to 10 for the models ADPE-XRC-6T and ADPE-XRC-6T-L.

Release 1.4.3

This release implements ADMXRC3 API Specification version 1.4.0.

Corrections:

1 Fixed a crash that can occur when attempting to do two or more DMA transfers on the same DMA channel.

2 Fixed a crash that can occur if the driver is somehow called by an admxrc3*.dll from a different driver version, due to the handlers for ADMXRC3_GetSensorInfo and ADMXRC3_ReadSensor failing to properly validate arguments.

3 Fixed an issue specific to the ADM-XRC-6T-ADV8 where the driver emitted the debug message "*** avrInit: failed to get AVR uC firmware version".

4 Added preliminary support for the models ADPE-XRC-6T and ADPE-XRC-6T-L.

Release 1.4.1

This release implements ADMXRC3 API Specification version 1.4.0.

Corrections:

1 Fixed a bug in the SI5338 clock synthesizer code for the ADM-XRC-6TGE that could corrupt memory when programming clock index 4. This clock generator is only available when the Si5338ExposeAllClocks driver parameter is nonzero; by default it is not available.

2 Support for ADM-XRC-6T-ADV8 is now feature-complete; added support for programming VPD and reading system monitor sensors via ADMXRC3 API.

Release 1.4.0

This release implements ADMXRC3 API Specification version 1.4.0.

Enhancements:

1 Added new API functions for performing DMA transfer to arbitrary PCI-E addresses: ADMXRC3_ReadDMABus, ADMXRC3_StartReadDMABus, ADMXRC3_StartWriteDMABus, ADMXRC3_WriteDMABus.

2 Added caching mechanism for Flash memory; reduces delays in execution of Flash API functions when performing many small write and/or erase operations.

Release 1.3.1

This release implements ADMXRC3 API Specification version 1.3.0.

New behavior:

1 Added support for the ADM-XRC-6TGE.

2 Added preliminary support for the ADM-XRC-6T-ADV8.

3 For the ADM-XRC-6TL, now recognizes "Extended" temperature range value (2) in VPD at offset 0x3E.

4 For the ADM-XRC-6T1, now recognizes "Extended" temperature range value (2) in VPD at offset 0x42.

Enhancements:

5 Now exposes (via the ADMXRC3 API) a programmable clock generator with index 0 on the ADM-XRC-6T1 when it has firmware 1.6 (PCI revision 0x06) or later.

Corrections:

6 ADMXRC3_GetClockFrequency now correctly returns the current clock frequency for a given clock generator. The driver now interrogates the hardware at startup to determine the current frequencies generated by each clock generator, so that ADMXRC3_GetClockFrequency can return the correct frequency even before any call to ADMXRC3_SetClockFrequency.

7 ADMXRC3_GetClockFrequency now correctly validates the pointer argument (3rd argument) passed to it, and returns ADMXRC3_NULL_POINTER if it is NULL.

8 Corrected the maximum frequency allowed for the clock generator with index 0 on the ADM-XRC-6TL. Previously, the driver incorrectly permitted frequencies up to 210 MHz to be requested, whereas 140 MHz is the correct maximum frequency.

9 Fixed ADMXRC3_EraseFlash failing to correctly validate the region specified to ensure that it is wholly within the unprotected region of a Flash memory bank.

Release 1.2.0

This release implements ADMXRC3 API Specification version 1.2.0.

Corrections:

1 Fixed a problem in admxrc3.dll and admxrc3d.dll that could result in an application crash when a thread terminates.

Enhancements:

2 Added new API functions for performing DMA transfers with 64-bit local addresses:

- ADMXRC3_ReadDMAEx
- ADMXRC3_ReadDMALockedEx
- ADMXRC3_StartReadDMAEx
- ADMXRC3_StartReadDMALockedEx
- ADMXRC3_StartWriteDMAEx
- ADMXRC3_StartWriteDMALockedEx
- ADMXRC3_WriteDMAEx
- ADMXRC3_WriteDMALockedEx

3 Added support for new sensors in ADM-XRC-6TL and ADM-XRC-6T1 with firmware 1.4 or later. This provides additional sensors that show internal temperature and voltages in the PCI Express to OCP Bridge.

Release 1.1.2

Corrections:

1 The EnableVpdWrite driver parameter is now written to the registry (with the value 0) when the driver is installed or reinstalled, to ensure that it has the correct default value.

2 Fixed a user-mode memory leak that can occur when the API functions ADMXRC3_LoadBitstreamA and ADMXRC3_LoadBitstreamW return a failure status code.

3 Fixed a bug in ADMXRC3_SetClockFrequency where on failure, the wrong status code was returned.

Release 1.1.0

Corrections:

1 Fixed a potential memory corruption issue on some architectures due to incorrect byte size calculation in device context structure.

2 Fixed a memory leak related to CFI Flash functionality when stopping and restarting the driver.

3 Fixed user VPD area (VPD space address range 0x100000-0x1FFFFF) not being accessible on ADM-XRC-6TL and ADM-XRC-6T1.

4 Corrected error codes returned by several ADMXRC3 API functions when invalid parameters are passed:

1 Non-Locked DMA functions now return 'ADMXRC3_INVALID_BUFFER' if the 'pBuffer' and/or 'length' parameters are invalid.

2 Locked DMA functions now return 'ADMXRC3_INVALID_BUFFER_HANDLE' if the 'hBuffer' parameter is invalid.

3 ADMXRC3_Unlock now returns 'ADMXRC3_INVALID_BUFFER_HANDLE' if the 'hBuffer' parameter is invalid.

5 Fixed 'ADMXRC3_UnregisterWin32Event' always failing with 'ADMXRC3_UNKNOWN_ERROR' when called from 32-bit process running on a 64-bit edition of Windows.

6 Fixed 'ADMXRC3_Unconfigure' failing with 'ADMXRC3_NOT_OWNER' even when the target FPGA has no owner.

7 Fixed the ADMXRC3 DMA functions not properly co-validating the 'local' and 'length' parameters.

8 Fixed the ADMXRC3 nonblocking DMA functions incorrectly returning 'ADMXRC3_PENDING' instead of an error code when bad parameters are passed.

9 Fixed the ADMXRC3 nonblocking DMA functions incorrectly returning 'ADMXRC3_SUCCESS' instead of 'ADMXRC3_PENDING' when 'length' is zero and all other parameters are OK.

10 Fixed the ADMXRC3 Locked DMA functions not properly co-validating the 'offset' and 'length' parameters.

11 Fixed certain ADMXRC3 API functions not trapping NULL pointers being passed, typically resulting in an application crash.

12 Implemented VPD write-protection mechanism (now protected by default).

13 Added workaround for 4k crossing issue in ADB3 Bridge rev 0x01 and earlier. Workaround is not applied for rev 0x02 or later.

Enhancements:

14 Added support for ADM-XRC-6T1.

15 Added API function ADMXRC3_OpenEx, which allows an unprivileged process to open a device in "read only" mode and call a subset of the ADMXRC3 API functions.

16 Added API functions ADMXRC3_StartNotificationWait and ADMXRC3_FinishNotificationWait, which allow Linux applications to wait for events (since there is no Linux equivalent of ADMXRC3_RegisterWin32Event).

17 Added API function ADMXRC3_GetCardInfoEx (with structure ADMXRC3_CARD_INFOEX), which returns a superset of data supplied by ADMXRC3_GetCardInfoEx.

18 Added diagnostic API functions ADMXRC3_GetSensorInfo and ADMXRC3_ReadSensor, with associated types and structures. These functions allow applications to monitor the health of a Gen 3 reconfigurable computing card.

Release 1.0.1

Corrections:

1 The correct default security attributes are now applied to devices, preventing non-privileged users from opening a device using the ADMXRC3 API. In release 1.0.0, a non-privileged user could open a device.

Release 1.0.0

This is the first release of the ADB3 Driver for Windows.

Address: 4 West Silvermills Lane, Edinburgh, EH3 5BD, UK
Telephone: +44 131 558 2600
Fax: +44 131 558 2700
email: sales@alpha-data.com
website: http://www.alpha-data.com

Address: 3507 Ringsby Court Suite 105, Denver, CO 80216
Telephone: (303) 954 8768
Fax: (866) 820 9956 toll free
email: sales@alpha-data.com
website: http://www.alpha-data.com