

SLAC AES Stream kernel driver Version 5.17.2 for the PCIe Xilinx KCU1500 FPGA Card

Installation on RedHawk Linux

RIQ 13276A-031

Release Notes

May 18th, 2022



This page intentionally blank

Table of Contents

1. INTRODUCTION	4
2. HARDWARE INSTALLATION.....	4
3. SOFTWARE INSTALLATION	5
3.1. Recompiling the <i>aes-stream</i> driver for other RedHawk kernels	5
4. LOADING THE <i>AES-STREAM</i> DRIVER	6
5. TESTING AND USAGE	7
6. REMOVING THE <i>AES-STREAM</i> DRIVER.....	9
7. NOTES AND ERRATA	10
8. PCIE KCU1500 CARD	11

1. Introduction

The KCU1500 data center board for the Xilinx® Kintex® UltraScale™ FPGA implements a Xilinx FPGA-based PCIe® accelerator add-in card for use in open compute project servers. This accelerator card is PCIe Gen 3 x16 compliant in a full-height half-length form factor, and can be used in servers that support PCIe x16 cards with bifurcation to dual x8 links or single x8 link without bifurcation. The accelerator card is designed for PCIe x16 bifurcated into two PCIe Gen3 x8 (or lower) links. PCIe bifurcation allows the card to enumerate as two independent PCIe links.

This document is intended to assist the user in installing Concurrent Real-Time's port of the SLAC **aes-stream** kernel driver to RedHawk Linux for the KCU1500 card.

This version of the **aes-stream** driver is valid for RedHawk Linux 7.5 only.

2. Hardware Installation

The KCU1500 FPGA card is a PCI Express x16 card compatible with most x16 PCI Express slots.



Caution: when installing an KCU1500 I/O card insure the computer is powered off and the machine's power cord is disconnected. Please observe electrostatic discharge precautions such as the use of a grounding strap.

After installing the card(s), reboot the system and check that the hardware has been recognized by the operating system by executing the following command:

```
# lspci -d 1A4A:
```

The **lspci** command will list all cards with the KCU1500 I/O Products assigned PCI Vendor ID of 0x1A4A found in the system:

```
82:00.0 Signal processing controller: SLAC National Accelerator Lab TID-AIR AXI
Stream DAQ PCIe card
```

The number of lines displayed by the **lspci** command should match the number of installed cards.

```
# lspci -d 1A4A: -v
```

```
82:00.0 Signal processing controller: SLAC National Accelerator Lab TID-AIR AXI
Stream DAQ PCIe card
Subsystem: SLAC National Accelerator Lab TID-AIR AXI Stream DAQ PCIe card
Physical Slot: 4
Flags: bus master, fast devsel, latency 0, IRQ 31, NUMA node 1
Memory at fa000000 (32-bit, non-prefetchable) [size=16M]
Capabilities: [80] Power Management version 3
Capabilities: [c0] Express Endpoint, MSI 00
Capabilities: [100] Advanced Error Reporting
Capabilities: [300] #19
Kernel driver in use: datadev
Kernel modules: bwtkdirvr, datadev
```

3. Software Installation

The *aes-stream* software is distributed on a DVD in RPM format. Included are source files for the provided software – the *aes-stream* device driver, vendor supplied access library, installation scripts, and a simple test program.

Caution: Before installing the software, the kernel build environment **must** be set up and match the current OS kernel you are using. If you are running one of the preconfigured kernels supplied by Concurrent and have not previously done so, run the following commands while logged in as the root user before installing the driver software:

```
# cd /lib/modules/$(uname -r)/build
# ./ccur-config -c -n
```

If you have built and are running a customized kernel configuration the kernel build environment should already have been set up when that custom kernel was built.

This requirement is enforced by the RPM package. If *ccur-config* has not been previously run on your system you will see an error message similar to the following when attempting to install the package:

```
#####
  Unsupported or un-configured kernel loaded. UTS_VERSION define does not
  exist in either files ...

  (the rest of the error message is not shown)
#####
```

The *aes-stream* kernel driver package is supplied in *rpm* format on a DVD. Log in as the root user and then insert the installation DVD into the drive. The system should auto-mount the DVD to a mount point in the */run/media/user* directory based on the DVD's volume label and the logged in user);

```
== as root ==
# cd /run/media/[user_name]/
# rpm -ivh aes-stream_driver/aes-stream*_RedHawk_driver_r*.rpm
# eject
```

Note: If the proper kernel build environment was not previously set up an error will occur during the installation of the RPM. Follow the instructions issued by the RPM command, or after correcting the error follow the steps in the [Recompiling the aes-stream driver for other RedHawk kernels](#) section to compile and install the driver.

3.1. Recompiling the *aes-stream* driver for other RedHawk kernels

RedHawk Linux™ is provided with three different kernels: the 'static', 'trace-static', and 'debug-static' variants. The *datadev.ko* kernel driver is initially built for the kernel variant or 'flavor' that was booted when this package was installed. The driver module will need to be rebuilt if operation with a different kernel flavor is desired.

To build and install just the *aes-stream* driver for another RedHawk Linux kernel flavor (say, the *-debug* kernel), perform the following steps. First, **reboot the system** to the desired kernel variant. Then:

```

== as root ==
# cd /lib/modules/$(uname -r)/build
# ./ccur-config -c -n          (set up the proper kernel build environment)
# cd /usr/local/CCRT/drivers/aes-stream
# make clean
# make                        (build kernel loadable module)
# make install                (install the new driver module)
# make load                    (loads the driver)

```

The **aes-stream** driver will also need to be rebuilt and reinstalled if a new RedHawk point release is installed – for example, if upgrading the kernel.

4. Loading the **aes-stream** driver

The **aes-stream** driver is a dynamically loadable driver. Once you install the **rpm** or perform the “make install”, appropriate installation files are placed in system directories so that the driver is automatically loaded and unloaded when Linux is booted or shutdown. If for any reason you do not wish to automatically load and unload the driver when Linux is booted or shutdown, or later want to again auto-load the driver, issue one of the following commands:

```

== as root ==

# /usr/bin/systemctl enable aes-stream (enable auto-loading of the driver)
# /usr/bin/systemctl disable aes-stream (disable auto-loading of the driver)

```

To manually load or unload the **aes-stream** kernel driver:

```

== as root ==

# /usr/bin/systemctl start aes-stream
# /usr/bin/systemctl stop aes-stream

-- or --

# cd /usr/local/CCRT/drivers/aes-stream
# make load
# make unload

```

If the **datadev.ko** driver is loaded successfully and sees the KCU1500 card then you should see a message on the system console or via the **dmesg** command similar to that below:

```

[56191.503580] datadev: Init
[56191.505041] (NULL device *): Init: Mapping Register space 0xfa000000 with size 0x1000000.
[56191.511969] (NULL device *): Init: Mapped to 0xffffc90008000000.
[56191.516685] datadev 0000:82:00.0: Init: Setting user reset
[56191.520882] datadev 0000:82:00.0: Init: Clearing user reset
[56191.525157] datadev 0000:82:00.0: Init: Using 40-bit DMA mask.
[56191.529702] datadev 0000:82:00.0: Init: Creating device class
[56191.534221] datadev 0000:82:00.0: Init: Creating 1024 TX Buffers. Size=131072 Bytes. Mode=1.
[56191.568594] datadev 0000:82:00.0: Init: Created 1024 out of 1024 TX Buffers. 134217728 Bytes.
[56191.575990] datadev 0000:82:00.0: Init: Creating 1024 RX Buffers. Size=131072 Bytes. Mode=1.
[56191.606121] datadev 0000:82:00.0: Init: Created 1024 out of 1024 RX Buffers. 134217728 Bytes.
[56191.613464] datadev 0000:82:00.0: Init: Read ring at: sw 0xffff880844510000 -> hw 0x844510000.
[56191.620863] datadev 0000:82:00.0: Init: Write ring at: sw 0xffff8808524c0000 -> hw 0x8524c0000.
[56191.628448] datadev 0000:82:00.0: Init: Found Version 2 Device. Desc128En=1
[56191.634113] datadev 0000:82:00.0: Init: IRQ 31
[56191.637360] datadev 0000:82:00.0: Init: Reg space mapped to 0xffffc90008000000.
[56191.643451] datadev 0000:82:00.0: Init: User space mapped to 0xffffc90008010000 with size 0xff0000.
[56191.651198] datadev 0000:82:00.0: Init: Top Register = 0x4011401

```

When the driver is unloaded, you should see a message similar to that below:

```
[65275.780526] datadev: Exit.
[65275.781958] datadev: Remove: Remove called.
[65275.786417] datadev 0000:82:00.0: Clean: Destroying device class
[65275.791226] datadev: Remove: Driver is unloaded.
```

The default driver options when loading the driver are in the `/etc/modprobe.d/datadev.conf` file.

```
# Copy this file to /etc/modprobe.d/datadev.conf and then edit to your specific needs

### Modified Configuration to use polling instead of interrupt
### options datadev cfgTxCount=1024 cfgRxCount=1024 cfgSize=131072 cfgMode=1 cfgCont=1
cfgIrqDis=1

### Original Configuration in zip file
options datadev cfgTxCount=1024 cfgRxCount=1024 cfgSize=131072 cfgMode=1 cfgCont=1
```

If you wish to change the driver configuration, simply edit this file and issue the command:

```
# /usr/bin/systemctl reload aes-stream
```

5. Testing and usage

The following commands can be used to test the `datadev` driver.

```
# cat /proc/datadev_0

----- Axi Version -----
Firmware Version : 0x1080200
ScratchPad : 0x0
Up Time Count : 56406
Device ID : 0x0
Git Hash : 5fb1bab0cf7be0ba0ba18515a586e1bc99dc55
DNA Value : 0x000000004002000101161de90d508245
Build String : XilinxKcu1500DmaLoopback: Vivado v2021.1, rdsrv317 (Ubuntu 20.04.3
LTS), Built Thu 28 Oct 2021 07:11:15 PM PDT by ruckman

----- General HW -----
Int Req Count : 0
Hw Dma Wr Index : 0
Sw Dma Wr Index : 0
Hw Dma Rd Index : 0
Sw Dma Rd Index : 0
Missed Wr Requests : 0
Missed IRQ Count : 20107
Continue Count : 0
Address Count : 4096
Hw Write Buff Count : 1024
Hw Read Buff Count : 0
Cache Config : 0x0
Desc 128 En : 1
Enable Ver : 0x4010a01
Driver Load Count : 10
IRQ Hold : 10000
BG Enable : 0x0

----- General -----
Dma Version : 0x6
Git Version :

----- Read Buffers -----
```

```

    Buffer Count : 1024
    Buffer Size : 131072
    Buffer Mode : 1
    Buffers In User : 0
    Buffers In Hw : 1024
    Buffers In Pre-Hw Q : 0
    Buffers In Rx Queue : 0
    Missing Buffers : 0
    Min Buffer Use : 0
    Max Buffer Use : 0
    Avg Buffer Use : 0
    Tot Buffer Use : 0

```

```

----- Write Buffers -----
    Buffer Count : 1024
    Buffer Size : 131072
    Buffer Mode : 1
    Buffers In User : 0
    Buffers In Hw : 0
    Buffers In Pre-Hw Q : 0
    Buffers In Sw Queue : 1024
    Missing Buffers : 0
    Min Buffer Use : 0
    Max Buffer Use : 0
    Avg Buffer Use : 0
    Tot Buffer Use : 0

```

You can test the card using the ***dmaRate*** and ***dmaWrite*** tests as follows. No external connections are required for this test:

Open a terminal window and enter the following:

```

# cd /usr/local/CCRT/drivers/aes-stream/aes-stream-drivers-main/data_dev/app/bin
# ./dmaRate --count=100000

```

Open a second terminal window and enter the following:

```

# cd /usr/local/CCRT/drivers/aes-stream/aes-stream-drivers-main/data_dev/app/bin
# ./dmaWrite 0 --count=1000000

```

You should see in the second window the following:

```

Write ret=1000, Dest=0, count=1
Write ret=1000, Dest=0, count=2
Write ret=1000, Dest=0, count=3
Write ret=1000, Dest=0, count=4
Write ret=1000, Dest=0, count=5
.
.
.
Write ret=1000, Dest=0, count=999996
Write ret=1000, Dest=0, count=999997
Write ret=1000, Dest=0, count=999998
Write ret=1000, Dest=0, count=999999
Write ret=1000, Dest=0, count=1000000

```

You should also see in the first window something similar to the following:

maxCnt	size	count	duration	rate	bw	Read uS	Return uS
--------	------	-------	----------	------	----	---------	-----------

1	1.000e+03	100000	5.96e+01	1.68e+03	1.34e+07	6	0
5	1.000e+03	100000	1.95e+00	5.13e+04	4.11e+08	6	1
1	1.000e+03	100000	6.99e+00	1.43e+04	1.14e+08	6	1
4	1.000e+03	100000	3.24e+00	3.09e+04	2.47e+08	5	1
2	1.000e+03	100000	3.19e+00	3.13e+04	2.51e+08	6	1
3	1.000e+03	100000	5.75e+00	1.74e+04	1.39e+08	6	1
3	1.000e+03	100000	3.59e+00	2.78e+04	2.23e+08	5	1
4	1.000e+03	100000	3.00e+00	3.34e+04	2.67e+08	5	1
1	1.000e+03	100000	5.85e+00	1.71e+04	1.37e+08	5	1

You can also enable/disable debug information in the driver as follows:

```
# cd /usr/local/CCRT/drivers/aes-stream/aes-stream-drivers-main/data_dev/app/bin
# ./setDebug 1
```

If you issue the **dmesg** command you should see something similar to being repeated every 3 to 5 microseconds:

```
[57632.374479] datadev 0000:82:00.0: Irq: Called.
[57632.377648] datadev 0000:82:00.0: Service: Entered
[57632.381156] datadev 0000:82:00.0: Service: Done. Handled = 0
```

To disable debugging do the following:

```
# cd /usr/local/CCRT/drivers/aes-stream/aes-stream-drivers-main/data_dev/app/bin
# ./setDebug 0
```

6. Removing the *aes-stream* driver

The ***aes-stream*** driver is a dynamically loadable driver that can be unloaded, uninstalled, and removed. Once removed, the only way to recover the driver is to re-install the ***rpm*** from the installation media.

If for any reason, the user wishes to simply unload the driver from the running kernel, they may perform one of the following:

```
=== as root ===
# /usr/bin/systemctl stop aes-stream
    --- or ---
# cd /usr/local/CCRT/drivers/aes-stream
# make unload      (unload the driver from the kernel)
```

To uninstall the ***aes-stream*** driver and its support files from the system directories, issue the following commands:

```
=== as root ===
# cd /usr/local/CCRT/drivers/aes-stream
# make uninstall      (uninstall the kernel driver and supporting files)
```

In this way, the user can simply issue the “make install” and “make load” in the ***/usr/local/CCRT/drivers/aes-stream*** directory at a later date to re-install and re-load the driver.

CAUTION: If any local changes or modifications to the ***aes-stream*** driver package, they should be saved prior to invoking the following rpm remove command or else they will be lost.

To completely remove the **aes-stream** driver package from the system, enter the following **rpm** command.

```
=== as root ===  
# rpm --erase aes-stream (driver unloaded, uninstalled, and deleted)
```

7. Notes and Errata

This package is installable only on RedHawk CentOS Linux releases 7.5

The **aes-stream** kernel package supports many I/O products. However, only the **PCIe-KCU1500 card** has been tested.

Several driver packages require cross-compile files that are local to SLAC. These drivers are unable to be compiled or tested by Concurrent.

It should be noted the **aes-stream** driver continuously generates interrupts when the interrupt option is configured (i.e. `cfgIrqDis=0`). This can be easily observed by running the `setDebug` test with an argument of 1 and issuing the 'dmesg' command. You can disable the debug by setting the argument to 0.

8. PCIe KCU1500 Card



This page intentionally blank