Xilinx Answer 72723
QDMA Linux Kernel Driver Usage and Debug Guide

Important Note: This downloadable PDF of an Answer Record is provided to enhance its usability and readability. It is important to note that Answer Records are Web-based content that are frequently updated as new information becomes available. You are reminded to visit the Xilinx Technical Support Website and review (Xilinx Answer 72723) for the latest version of this Answer Record.

Introduction

The example provided in this document uses a Xilinx Virtex UltraScale+ FPGA VCU118 Evaluation Kit, a Linux machine running CentOS (7.4.1708), Vivado 2019.1 and the QDMA Linux reference drivers 2019.1 release from Github.com

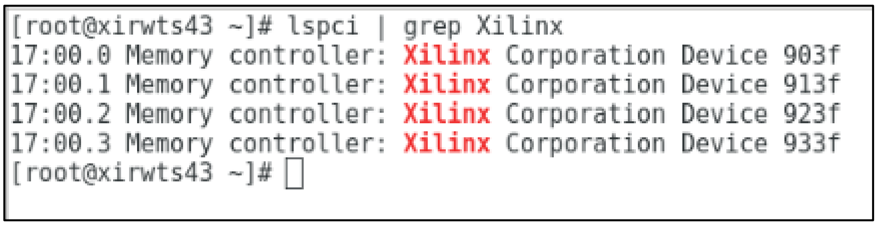
This document provides instructions for installing and running the QDMA Linux driver and the associated tools to configure and test the driver; ***it should be used in conjunction with the 'read me' and the documentation that comes with the driver.***

Device Detection

Before installing the drivers, run the following command to ensure that the Xilinx PCIe device is detected.

- lspci |grep Xilinx

The successful execution of the command should return something similar to what is shown below:



```
[root@xirwts43 ~]# lspci | grep Xilinx
17:00.0 Memory controller: Xilinx Corporation Device 903f
17:00.1 Memory controller: Xilinx Corporation Device 913f
17:00.2 Memory controller: Xilinx Corporation Device 923f
17:00.3 Memory controller: Xilinx Corporation Device 933f
[root@xirwts43 ~]# 
```

Figure 1 - List of Xilinx PCI connections

All of the devices shown in the screenshot are on bus 17. The next number 00 is the device number. The final number is the function number.

These numbers are important, and they will be needed during the use of the Linux driver. The numbers are referred to as the bus device function number of the Xilinx device, or BDF number. It is often written in the form of bbddf, which is each number without spaces, so in this case it would be 17000 for the first device and 17001,17002 and 17003 for the others respectively.

Installing the drivers and modules

Before the driver can be compiled, installed or configured, the libaio library needs to be installed. This can be done by running the following command:
Red Hat system:

- sudo yum install libaio1

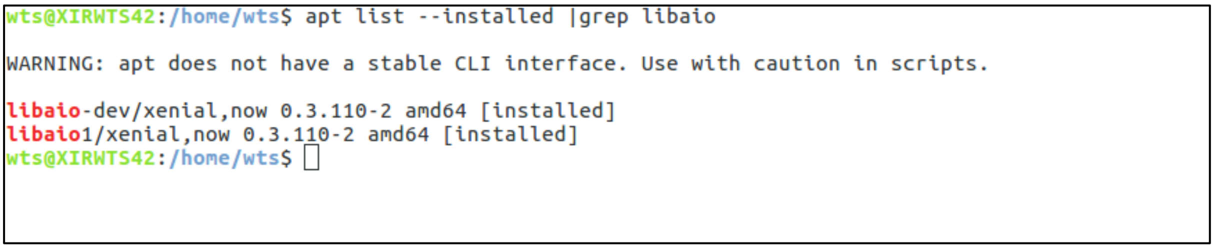
Ubuntu system:

- sudo apt-get install libaio libaio-devel

The command apt list can be used to list installed packages. However, as this will give a very long list, it is better to use grep to search for libaio.

The following command can be used

- apt list –installed |grep libaio



```
wts@XIRWTS42:/home/wts$ apt list --installed |grep libaio

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

libaio-dev/xenial,now 0.3.110-2 amd64 [installed]
libaio1/xenial,now 0.3.110-2 amd64 [installed]
wts@XIRWTS42:/home/wts$ 
```

Figure 2 - showing libaio is installed

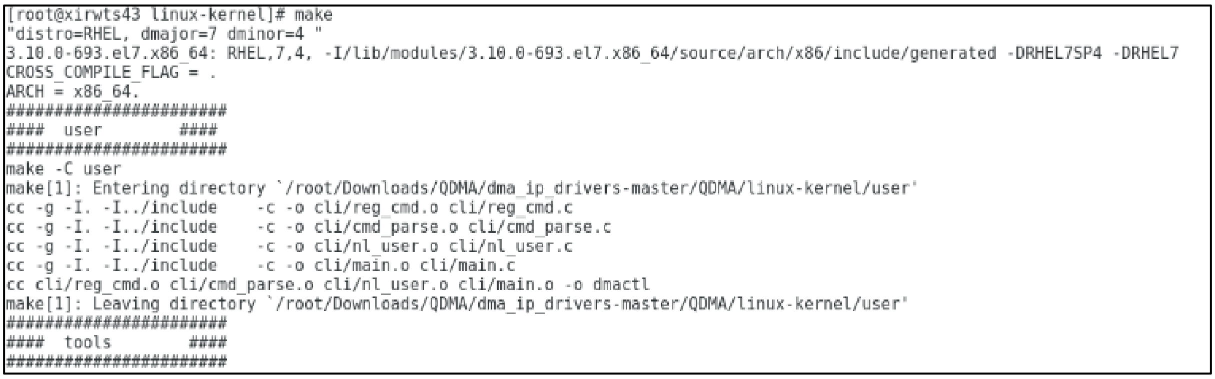
Once this header file is installed, the QDMA driver installation can proceed.

Once the QDMA Linux reference drivers 2019.1 release from Github.com has been downloaded, navigate to the following folder:

- dma_ip_drivers-master\QDMA\linux-kernel\

Once there, with root permission, run the following command:

- make

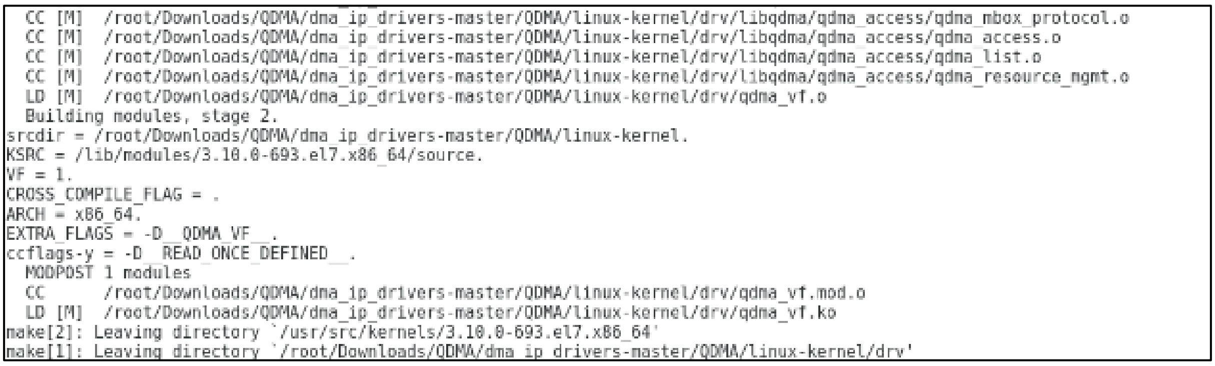


```
[root@xirwts43 linux-kernel]# make
"distro=RHEL, dmajor=7 dminor=4 "
3.10.0-693.el7.x86_64: RHEL,7,4, -I/lib/modules/3.10.0-693.el7.x86_64/source/arch/x86/include/generated -DRHEL7SP4 -DRHEL7
CROSS_COMPILE_FLAG = .
ARCH = x86_64.
#######################
####  user        ####
#######################
make -C user
make[1]: Entering directory `/root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/user'
cc -g -I. -I../include    -c -o cli/reg_cmd.o cli/reg_cmd.c
cc -g -I. -I../include    -c -o cli/cmd_parse.o cli/cmd_parse.c
cc -g -I. -I../include    -c -o cli/nl_user.o cli/nl_user.c
cc -g -I. -I../include    -c -o cli/main.o cli/main.c
cc cli/reg_cmd.o cli/cmd_parse.o cli/nl_user.o cli/main.o -o dmactl
make[1]: Leaving directory `/root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/user'
#######################
####  tools        ####
#######################
```

Figure 3 - Beginning of make command executing

Note: see the Appendix section for the full list of commands that the make command executes.

The last commands executed can be seen in the image below.

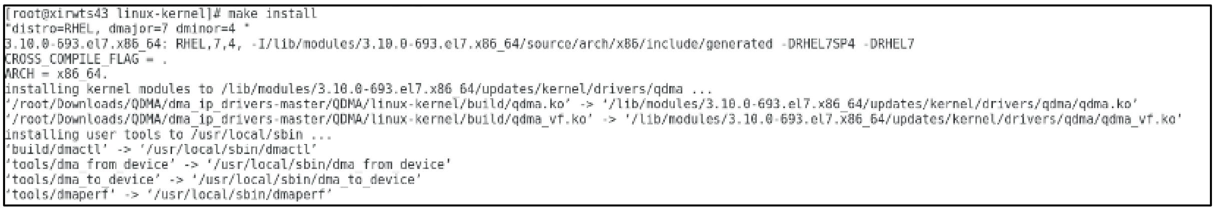


```
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_access/qdma_mbox_protocol.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_access/qdma_access.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_access/qdma_list.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_access/qdma_resource_mgmt.o
  LD [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/qdma_vf.o
  Building modules, stage 2.
srcdir = /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel.
KSRC = /lib/modules/3.10.0-693.el7.x86_64/source.
VF = 1.
CROSS_COMPILE_FLAG = .
ARCH = x86_64.
EXTRA_FLAGS = -D__QDMA_VF__.
ccflags-y = -D__READ_ONCE_DEFINED__.
  MODPOST 1 modules
  CC      /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/qdma_vf.mod.o
  LD [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/qdma_vf.ko
make[2]: Leaving directory `/usr/src/kernels/3.10.0-693.el7.x86_64'
make[1]: Leaving directory `/root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv'
```

Figure 4 - End of make command

Once the make command is finished executing as shown, run “make install”.

- make install

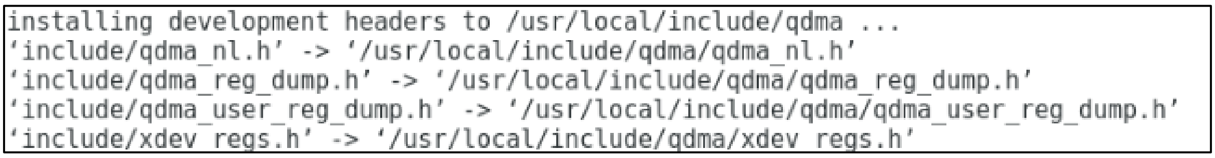


```
[root@xirwts43 linux-kernel]# make install
"distro=RHEL, dmajor=7 dminor=4 "
3.10.0-693.el7.x86_64: RHEL,7,4, -I/lib/modules/3.10.0-693.el7.x86_64/source/arch/x86/include/generated -DRHEL7SP4 -DRHEL7
CROSS_COMPILE_FLAG = .
ARCH = x86_64.
installing kernel modules to /lib/modules/3.10.0-693.el7.x86_64/updates/kernel/drivers/qdma ...
'/root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/build/qdma.ko' -> '/lib/modules/3.10.0-693.el7.x86_64/updates/kernel/drivers/qdma/qdma.ko'
'/root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/build/qdma_vf.ko' -> '/lib/modules/3.10.0-693.el7.x86_64/updates/kernel/drivers/qdma/qdma_vf.ko'
installing user tools to /usr/local/sbin ...
'build/dmactl' -> '/usr/local/sbin/dmactl'
'tools/dma_from_device' -> '/usr/local/sbin/dma_from_device'
'tools/dma_to_device' -> '/usr/local/sbin/dma_to_device'
'tools/dmaperf' -> '/usr/local/sbin/dmaperf'
```

Figure 5 - Beginning of make install command

Note: see the Appendix section for the full list of commands that the make install command executes.

The last commands executed by “make install” are shown below:



```
installing development headers to /usr/local/include/qdma ...
'include/qdma_nl.h' -> '/usr/local/include/qdma/qdma_nl.h'
'include/qdma_reg_dump.h' -> '/usr/local/include/qdma/qdma_reg_dump.h'
'include/qdma_user_reg_dump.h' -> '/usr/local/include/qdma/qdma_user_reg_dump.h'
'include/xdev_regs.h' -> '/usr/local/include/qdma/xdev_regs.h'
```

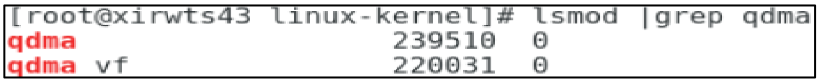
Figure 6 - End of make install command

After this command has been entered, the qdma.ko and the qdma_vf.ko modules should be installed to the following locations

- ‘/root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/build/qdma.ko’ -> ‘/lib/modules/3.10.0-693.el7.x86_64/updates/kernel/drivers/qdma/qdma.ko’
- ‘/root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/build/qdma_vf.ko’ -> ‘/lib/modules/3.10.0-693.el7.x86_64/updates/kernel/drivers/qdma/qdma_vf.ko’

To make sure that the modules were installed, run the following command:

- lsmod | grep qdma



```
[root@xirwts43 linux-kernel]# lsmod |grep qdma
qdma                   239510  0
qdma_vf                220031  0
```

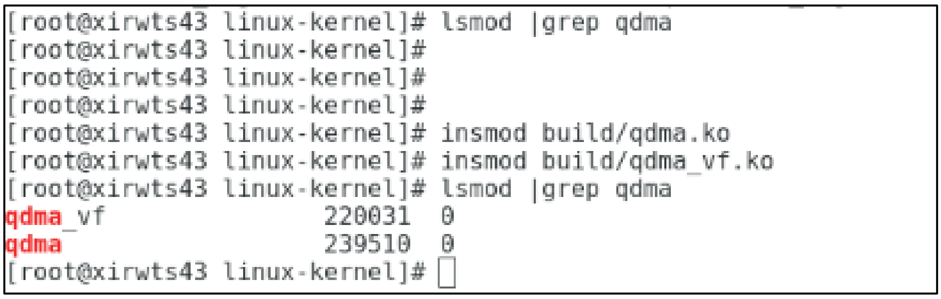
Figure 7 - List of QDMA modules

If there is no result when you search the list of modules for the string qdma, navigate to either of the modules in the build folder in the linux-kernal directory and install the module with the following command.

- insmod qdma.ko

To install the VF driver, run the following command:

- insmod build/qdma_vf.ko



```
[root@xirwts43 linux-kernel]# lsmod |grep qdma
[root@xirwts43 linux-kernel]#
[root@xirwts43 linux-kernel]#
[root@xirwts43 linux-kernel]#
[root@xirwts43 linux-kernel]# insmod build/qdma.ko
[root@xirwts43 linux-kernel]# insmod build/qdma_vf.ko
[root@xirwts43 linux-kernel]# lsmod |grep qdma
qdma_vf              220031  0
qdma                 239510  0
[root@xirwts43 linux-kernel]# 
```

Figure 8 - Installing the qdma modules

dmactl

The dmactl tool is installed in the following location by the “make install” command.

- /usr/local/sbin/dmactl

The following command can be run to bring up the manual in the terminal.

- man dmactl

Note: see the Appendix section for the 'man dmactl' output.

The following command provides an overview of its usage.

- dmactl -h

Note: see the Appendix section for the 'dmactl -h' output.

The dmactl tool can be used for the following:

a. To list, add, start, stop and delete queues on QDMA devices.
b. Read and write to a register on the device and dump the configuration bar and user bar registers
c. Display a Queue’s configuration parameters, its descriptor ring entries, its completion ring entries and its interrupt ring entries.

dma_to_device

The dma_to_device is installed to the following location by the make install command.

- /usr/local/sbin/dma_to_device

The following command provides an overview of its usage.

- dma_to_device -h

Note: see the Appendix section for 'dma_to_device -h' output.

dma_to_device is a user application tool provided along with QDMA Linux driver to perform the Host to Card data transfers.

dma_from_device

The dma_from_device tool is installed to the following location by the make install command.

- /usr/local/sbin/dma_from_device

The following command provides an overview of its usage.

- dma_from_device -h

Note: see the Appendix section for 'dma_from_device -h' output.

dma_from_device is a user application tool provided along with the QDMA Linux driver to perform the Card to Host data transfers.

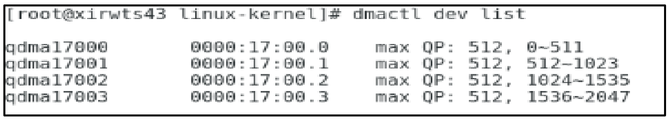
Configuring the QDMA Driver

Step 1: Searching for QDMA devices

The first step in configuring the driver is to use the Xilinx QDMA tool, dmactl, to search for all QDMA configured devices connected to the machine. This can be done by running the following command.

- dmactl dev list

When the dmactl dev list command is entered and the following result is returned, and if the maximum number of queues for any of the devices does not need to be changed, proceed to step 3.

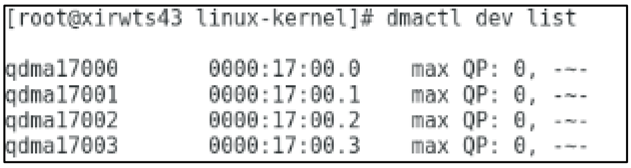


```
[root@xirwts43 linux-kernel]# dmactl dev list

qdma17000       0000:17:00.0    max QP: 512, 0~511
qdma17001       0000:17:00.1    max QP: 512, 512~1023
qdma17002       0000:17:00.2    max QP: 512, 1024~1535
qdma17003       0000:17:00.3    max QP: 512, 1536~2047
```

Figure 9 - QDMA control tool list of qdma devices

This image shows that each device has 512 maximum queues. The queues start at index 0 and go up to index 2047, for a maximum number of 2048 of queues.

If the list of devices shows that each device has a maximum number of queues of 0, as shown below, go to step 2 to change this.



```
[root@xirwts43 linux-kernel]# dmactl dev list

qdma17000       0000:17:00.0    max QP: 0, -~-
qdma17001       0000:17:00.1    max QP: 0, -~-
qdma17002       0000:17:00.2    max QP: 0, -~-
qdma17003       0000:17:00.3    max QP: 0, -~-
```

Figure 10 - dmactl, maximum number of queues is zero

Step 2: Configuring the maximum number of queues

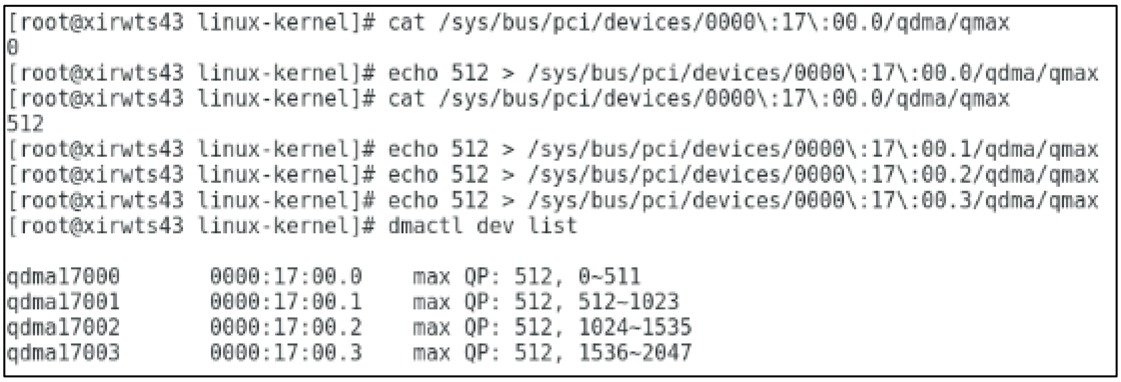
The file that needs to be changed to set the maximum number of queues can be found at */sys/bus.pci/devices/0000\:bb\:dd.f/qdma/qmax*. Where bb is the bus number, dd, the device number and f is the function number.

The cat command is used to concatenate and print the output of the file in the terminal. This can be done by using the following command:

- cat /sys/bus/pci/devices/0000\:<bb>\:<dd>.<f>/qdma/qmax

To change the maximum number of queues, use the echo command, and enter a number up to 2048. As long as the total number of queues for the device does not exceed 2048, it can be set so that one device has all of the queues, only half of the queues, or to spread them out. This can be done as follows

- echo 512 > /sys/bus/pci/devices/0000\:<bb>\:<dd>.<f>/qdma/qmax



```
[root@xirwts43 linux-kernel]# cat /sys/bus/pci/devices/0000\:17\:00.0/qdma/qmax
0
[root@xirwts43 linux-kernel]# echo 512 > /sys/bus/pci/devices/0000\:17\:00.0/qdma/qmax
[root@xirwts43 linux-kernel]# cat /sys/bus/pci/devices/0000\:17\:00.0/qdma/qmax
512
[root@xirwts43 linux-kernel]# echo 512 > /sys/bus/pci/devices/0000\:17\:00.1/qdma/qmax
[root@xirwts43 linux-kernel]# echo 512 > /sys/bus/pci/devices/0000\:17\:00.2/qdma/qmax
[root@xirwts43 linux-kernel]# echo 512 > /sys/bus/pci/devices/0000\:17\:00.3/qdma/qmax
[root@xirwts43 linux-kernel]# dmactl dev list

qdma17000       0000:17:00.0    max QP: 512, 0~511
qdma17001       0000:17:00.1    max QP: 512, 512~1023
qdma17002       0000:17:00.2    max QP: 512, 1024~1535
qdma17003       0000:17:00.3    max QP: 512, 1536~2047
```

Figure 11 - Print the maximum number of Queues and change the maximum number of Queues

If the maximum number of queues cannot be changed, check each device to see if any queues are configured. If so stop and then delete them.

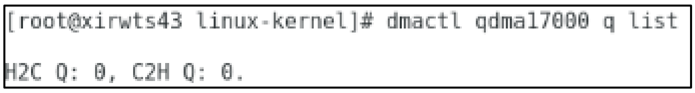
Configuring the Queues

Step 1 : Listing the configured queues

Use the following command to display the number of queues and if they have been started.

- dmactl qdma<bbddf> q list

If there are no queues configured, something similar to the following should be seen.

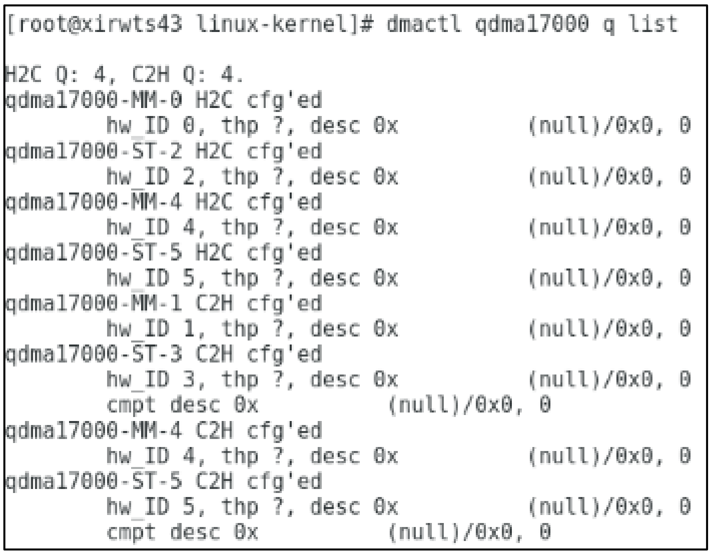


```
[root@xirwts43 linux-kernel]# dmactl qdma17000 q list

H2C Q: 0, C2H Q: 0.
```

Figure 12 - List of configured Queues

H2C Q refers to the number of host to card queues and C2H Q refers to the number of card to host queues. Proceed to step 2 to add a queue.

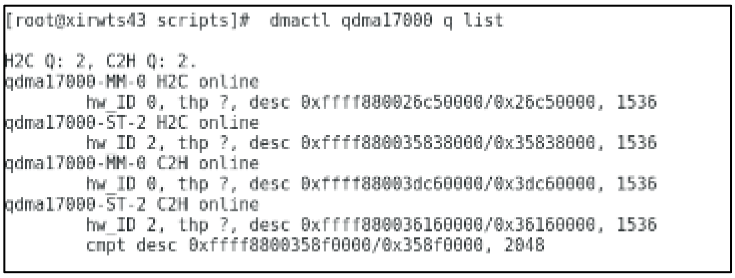
If there are queues configured and it looks like the image shown below, it means that there are queues configured but no descriptors yet, which means that the queues have not been started. The MM and ST refer to either a memory mapped channel or a streaming channel respectively. If the direction of the queue is bidirectional, the index will show up under both H2C and C2H.



```
[root@xirwts43 linux-kernel]# dmactl qdma17000 q list

H2C Q: 4, C2H Q: 4.
qdma17000-MM-0 H2C cfg'ed
        hw_ID 0, thp ?, desc 0x          (null)/0x0, 0
qdma17000-ST-2 H2C cfg'ed
        hw_ID 2, thp ?, desc 0x          (null)/0x0, 0
qdma17000-MM-4 H2C cfg'ed
        hw_ID 4, thp ?, desc 0x          (null)/0x0, 0
qdma17000-ST-5 H2C cfg'ed
        hw_ID 5, thp ?, desc 0x          (null)/0x0, 0
qdma17000-MM-1 C2H cfg'ed
        hw_ID 1, thp ?, desc 0x          (null)/0x0, 0
qdma17000-ST-3 C2H cfg'ed
        hw_ID 3, thp ?, desc 0x          (null)/0x0, 0
        cmpt desc 0x          (null)/0x0, 0
qdma17000-MM-4 C2H cfg'ed
        hw_ID 4, thp ?, desc 0x          (null)/0x0, 0
qdma17000-ST-5 C2H cfg'ed
        hw_ID 5, thp ?, desc 0x          (null)/0x0, 0
        cmpt desc 0x          (null)/0x0, 0
```

Figure 13 - List of added stopped Queues

If there are started queues, the descriptors of the queues will be configured as shown in the following image.



```
[root@xirwts43 scripts]#  dmactl qdma17000 q list

H2C Q: 2, C2H Q: 2.
qdma17000-MM-0 H2C online
        hw_ID 0, thp ?, desc 0xffff880026c50000/0x26c50000, 1536
qdma17000-ST-2 H2C online
        hw_ID 2, thp ?, desc 0xffff880035838000/0x35838000, 1536
qdma17000-MM-0 C2H online
        hw_ID 0, thp ?, desc 0xffff88003dc60000/0x3dc60000, 1536
qdma17000-ST-2 C2H online
        hw_ID 2, thp ?, desc 0xffff880036160000/0x36160000, 1536
        cmpt desc 0xffff8800358f0000/0x358f0000, 2048
```

Figure 14 - List of started Queues

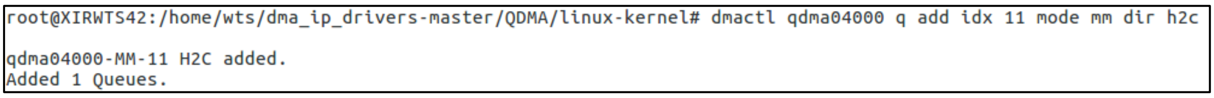
Step 2 : Adding a Queue

To add a queue, the following needs to be specified:

a) The device the queue is to be added to.
b) The index of the queue that it is to be made at (needs to be within the range of the maximum number of queues, for that device).
c) The descriptor mode of the channel that is to be added, in lowercase, either mm or st , for memory mapped or streaming (defaults to mm).
d) The direction of the queue, either h2c, c2h or bi, for host to card, card to host or bidirectional (defaults to h2c) respectively.

The command is shown below,

- dmactl qdma<bbddf> q add idx 0 mode mm dir h2c



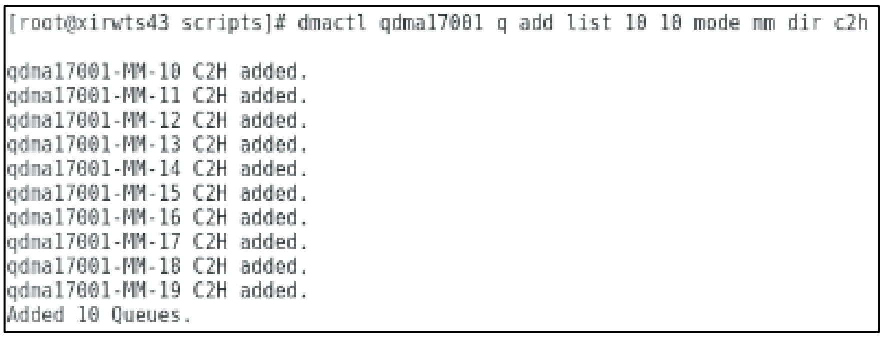
```
root@XIRWTS42:/home/wts/dma_ip_drivers-master/QDMA/linux-kernel# dmactl qdma04000 q add idx 11 mode mm dir h2c

qdma04000-MM-11 H2C added.
Added 1 Queues.
```

Figure 15 - Adding a queue to a device

A list of Queues can also be added of the same type at the same time. This can be done with the following command:

- dmactl qdma<bbddf> q add list 10 10 mode mm dir c2h

This command adds a list of 10 queues starting at index 10 in MM mode in the card to host direction; each queue will be of the same mode and direction.



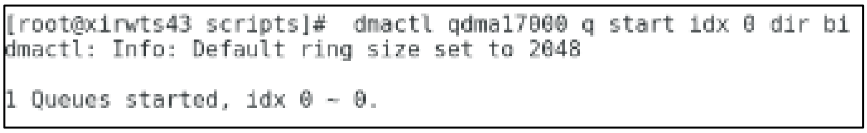
```
[root@xirwts43 scripts]# dmactl qdma17001 q add list 10 10 mode mm dir c2h

qdma17001-MM-10 C2H added.
qdma17001-MM-11 C2H added.
qdma17001-MM-12 C2H added.
qdma17001-MM-13 C2H added.
qdma17001-MM-14 C2H added.
qdma17001-MM-15 C2H added.
qdma17001-MM-16 C2H added.
qdma17001-MM-17 C2H added.
qdma17001-MM-18 C2H added.
qdma17001-MM-19 C2H added.
Added 10 Queues.
```

Figure 16 - Adding 10 card to host queues starting at index 10

Step 3 : Starting a Queue

To start a queue, the dmactl tool is used. The QDMA device, the index and the direction of the queue, either h2c, c2h or bi, needs to be specified. This command can be seen below:

- dmactl qdma<bbddf> q start idx 0 dir bi



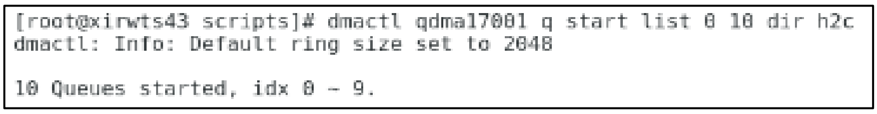
```
[root@xirwts43 scripts]#  dmactl qdma17000 q start idx 0 dir bi
dmactl: Info: Default ring size set to 2048

1 Queues started, idx 0 ~ 0.
```

Figure 17 - Starting a queue

A list of queues can also be started at the same time, these queues must have the same direction, but they can have different modes. This command is shown below.

- dmactl qdma<bbddf> q start list 0 10 dir h2c

This command starts a list of 10 queues, starting with index 0, with host to card direction.



```
[root@xirwts43 scripts]# dmactl qdma17001 q start list 0 10 dir h2c
dmactl: Info: Default ring size set to 2048

10 Queues started, idx 0 ~ 9.
```

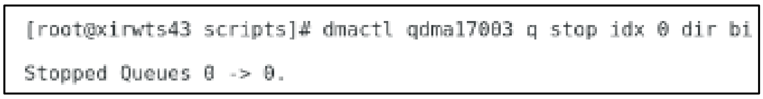
Figure 18 - Starting a list of queues

Using the start queue command, several different parameters can be set. See the Appendix section for the list of these parameters. These include prefetch enable, enable descriptor bypass, and enable completion status. An explanation of what these modes are used for can be found in the QDMA Product Guide.

Step 4 : Stopping a Queue

To stop a queue, the dmactl tool is used. The QDMA device, the index and the direction of the queue, either h2c, c2h or bi, needs to be specified. The mode is not considered for this command. This command can be seen below.

- dmactl qdma<bbddf> q stop idx 0 dir h2c



```
[root@xirwts43 scripts]# dmactl qdma17003 q stop idx 0 dir bi
Stopped Queues 0 -> 0.
```

Figure 19 - Stopping a Queue

A list of queues can also be stopped, this command can be seen below:

- dmactl qdma<bbddf> q stop list 0 10 dir h2c

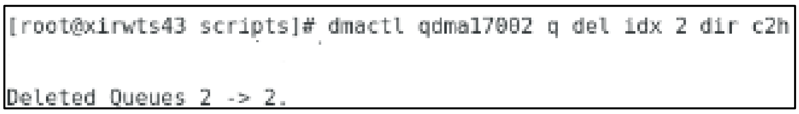
This command stops a list of 10 queues, starting with index 0, with a host to card direction.

Step 5 : Deleting a Queue

To delete a queue, the dmactl tool is used. The QDMA device, the index and the direction of the queue, either h2c, c2h or bi, needs to be specified. The mode is not considered for this command. This command is shown below:

- dmactl qdma<bbddf> q del idx 0 dir bi

This command stops a queue index 0, that is bidirectional:



```
[root@xirwts43 scripts]# dmactl qdma17002 q del idx 2 dir c2h

Deleted Queues 2 -> 2.
```

Figure 20 - Deleting a Queue

A list of queues can also be deleted:

- dmactl qdma<bbddf> q del list 0 10 dir bi

This command deletes a list of 10 Queues starting at index 0, that is bidirectional.

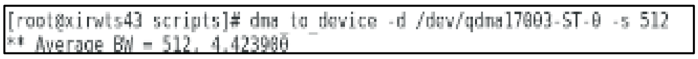
Usage and Testing of Configured Queues

Once queues have been started, dma_to_device and dma_from_device can be used.
The dma_to_device can be used to test host to card while the dma_from_device tool can be used to test card to host.

Writing to the Device

- dma_to_device -d /dev/qdma1<bbddf>-<ST||MM>-<index> -s 512

The -s command specifies the size, in bytes that is to be written to the device. The command returns the average bandwidth, and the size defaults to 32 bytes.

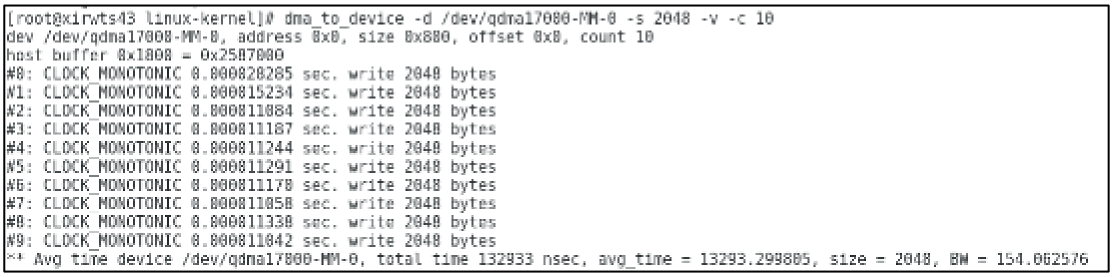


```
[root@xirwts43 scripts]# dma_to_device -d /dev/qdma17003-ST-0 -s 512
** Average BW = 512, 4.423980
```

Figure 21 - dma to device

If the -v command is used, it gives a verbose response. The -c command is used to specify the count i.e. the number of times the function will run to get a more accurate average bandwidth. -f is used to specify the file that is to be written to the device. The 16-bit pattern test file supplied in the scripts folder can be used to test for accuracy while writing to the device.

- dma_to_device -d /dev/qdma1<bbddf>-<ST||MM>-<index> -s 2048 -v -c 10



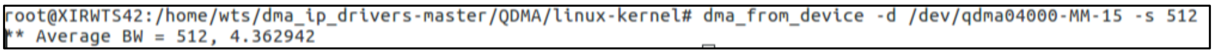
```
[root@xirwts43 linux-kernel]# dma_to_device -d /dev/qdma17000-MM-0 -s 2048 -v -c 10
dev /dev/qdma17000-MM-0, address 0x0, size 0x800, offset 0x0, count 10
host buffer 0x1000 = 0x2587000
#0: CLOCK_MONOTONIC 0.000028285 sec. write 2048 bytes
#1: CLOCK_MONOTONIC 0.000015234 sec. write 2048 bytes
#2: CLOCK_MONOTONIC 0.000011084 sec. write 2048 bytes
#3: CLOCK_MONOTONIC 0.000011187 sec. write 2048 bytes
#4: CLOCK_MONOTONIC 0.000011244 sec. write 2048 bytes
#5: CLOCK_MONOTONIC 0.000011291 sec. write 2048 bytes
#6: CLOCK_MONOTONIC 0.000011170 sec. write 2048 bytes
#7: CLOCK_MONOTONIC 0.000011058 sec. write 2048 bytes
#8: CLOCK_MONOTONIC 0.000011338 sec. write 2048 bytes
#9: CLOCK_MONOTONIC 0.000011042 sec. write 2048 bytes
** Avg time device /dev/qdma17000-MM-0, total time 132933 nsec, avg_time = 13293.299805, size = 2048, BW = 154.062576
```

Figure 22 - dma to device verbose, with count

Writing from the Device

- dma_from_device -d /dev/qdma<bbddf>-<ST||MM>-<index> -s 512

The -s command specifies the size, in bytes that is to be written from the device. The command returns the average bandwidth, the size defaults to 32 bytes. -v gives a verbose response.

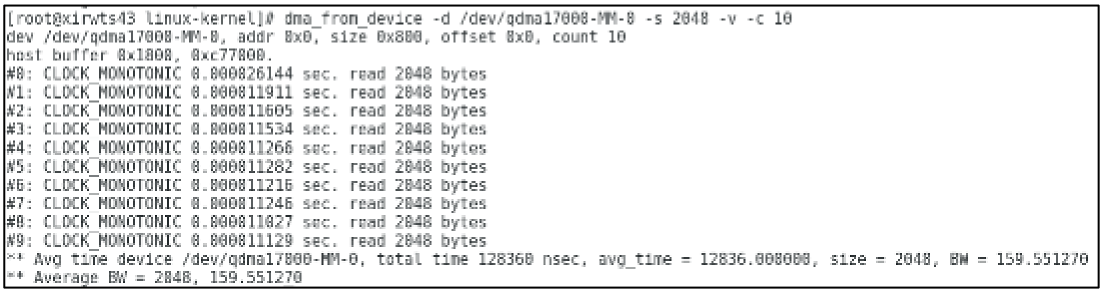


```
root@XIRWTS42:/home/wts/dma_ip_drivers-master/QDMA/linux-kernel# dma_from_device -d /dev/qdma04000-MM-15 -s 512
** Average BW = 512, 4.362942
```

Figure 23 - dma from device , verbose

If the -v command is used, it gives a verbose response. If the -c command is used, this is used to specify the count i.e. the number of times the function will run, to get a more accurate average bandwidth.

- dma_from_device -d /dev/qdma1<bbddf>-<ST||MM>-<index> -s 2048 -v -c 10



```
[root@xirwts43 linux-kernel]# dma_from_device -d /dev/qdma17000-MM-0 -s 2048 -v -c 10
dev /dev/qdma17000-MM-0, addr 0x0, size 0x800, offset 0x0, count 10
host buffer 0x1000, 0xc77000.
#0: CLOCK_MONOTONIC 0.000026144 sec. read 2048 bytes
#1: CLOCK_MONOTONIC 0.000011911 sec. read 2048 bytes
#2: CLOCK_MONOTONIC 0.000011605 sec. read 2048 bytes
#3: CLOCK_MONOTONIC 0.000011534 sec. read 2048 bytes
#4: CLOCK_MONOTONIC 0.000011266 sec. read 2048 bytes
#5: CLOCK_MONOTONIC 0.000011282 sec. read 2048 bytes
#6: CLOCK_MONOTONIC 0.000011216 sec. read 2048 bytes
#7: CLOCK_MONOTONIC 0.000011246 sec. read 2048 bytes
#8: CLOCK_MONOTONIC 0.000011027 sec. read 2048 bytes
#9: CLOCK_MONOTONIC 0.000011129 sec. read 2048 bytes
** Avg time device /dev/qdma17000-MM-0, total time 128360 nsec, avg_time = 12836.000000, size = 2048, BW = 159.551270
** Average BW = 2048, 159.551270
```

Figure 24 - dma from device verbose, with count

There is also a data generator which can be configured on the device so that it writes a 16 bit pattern to the host which can be tested against the supplied 16 bit pattern test file, in the scripts folder .-w is used to specify the file that is to be written to the host machine. The supplied 16-bit pattern test file, in the scripts folder, can be used to test for accuracy while writing to the device.

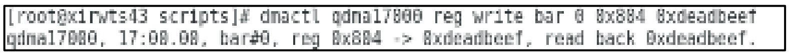
Reading, Writing and Dumping Registers

The dmactl tool can be used to read, write and dump the user and the configuration registers on the QDMA device. To read from or write to a device a Base Address Register (BAR) must be specified. The user BAR is 2, and the configuration BAR is 0. These commands can be seen below.

Writing to a register

The command below is used to write to the register 0x884 in the configuration BAR. If 0x is not put in front of the hex location of the register, as in 0x<Y>, where Y is the index number, the driver assumes that the input has been written in decimal and it will convert it to hex.

- dmactl qdma<bddff> reg write bar 0 0x884 0x<hexnumber>



```
[root@xirwts43 scripts]# dmactl qdma17000 reg write bar 0 0x884 0xdeadbeef
qdma17000, 17:00.00, bar#0, reg 0x884 -> 0xdeadbeef, read back 0xdeadbeef.
```

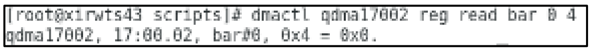
Figure 25 - Writing to a register

This command is useful to confirm the configuration of the driver. Further details can be found in QDMA Product Guide (PG302).

Reading from a register

The command below is used to read from the register 0x4 in the configuration BAR i.e. BAR 0

- dmactl qdma<bddff> reg read bar 0 0x4

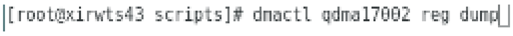


```
[root@xirwts43 scripts]# dmactl qdma17002 reg read bar 0 4
qdma17002, 17:00.02, bar#0, 0x4 = 0x0.
```

Figure 26 - Reading from a register

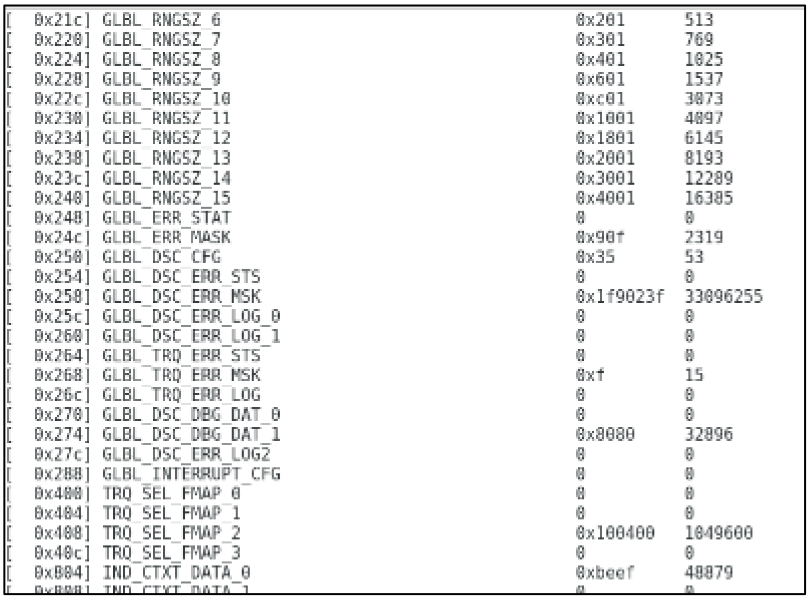
Dumping register information

The following command can be used to see the complete list of all registers and what is written into them.



```
[root@xirwts43 scripts]# dmactl qdma17002 reg dump
```

Figure 27 - Dumping of register information



```
[   0x21c] GLBL_RNGSZ_6                                          0x201      513
[   0x220] GLBL_RNGSZ_7                                          0x301      769
[   0x224] GLBL_RNGSZ_8                                          0x401      1025
[   0x228] GLBL_RNGSZ_9                                          0x601      1537
[   0x22c] GLBL_RNGSZ_10                                         0xc01      3073
[   0x230] GLBL_RNGSZ_11                                         0x1001     4097
[   0x234] GLBL_RNGSZ_12                                         0x1801     6145
[   0x238] GLBL_RNGSZ_13                                         0x2001     8193
[   0x23c] GLBL_RNGSZ_14                                         0x3001     12289
[   0x240] GLBL_RNGSZ_15                                         0x4001     16385
[   0x248] GLBL_ERR_STAT                                         0          0
[   0x24c] GLBL_ERR_MASK                                         0x90f      2319
[   0x250] GLBL_DSC_CFG                                          0x35       53
[   0x254] GLBL_DSC_ERR_STS                                      0          0
[   0x258] GLBL_DSC_ERR_MSK                                      0x1f9023f  33096255
[   0x25c] GLBL_DSC_ERR_LOG_0                                    0          0
[   0x260] GLBL_DSC_ERR_LOG_1                                    0          0
[   0x264] GLBL_TRQ_ERR_STS                                      0          0
[   0x268] GLBL_TRQ_ERR_MSK                                      0xf        15
[   0x26c] GLBL_TRQ_ERR_LOG                                      0          0
[   0x270] GLBL_DSC_DBG_DAT_0                                    0          0
[   0x274] GLBL_DSC_DBG_DAT_1                                    0x8080     32896
[   0x27c] GLBL_DSC_ERR_LOG2                                     0          0
[   0x288] GLBL_INTERRUPT_CFG                                    0          0
[   0x400] TRQ_SEL_FMAP_0                                        0          0
[   0x404] TRQ_SEL_FMAP_1                                        0          0
[   0x408] TRQ_SEL_FMAP_2                                        0x100400   1049600
[   0x40c] TRQ_SEL_FMAP_3                                        0          0
[   0x804] IND_CTXT_DATA_0                                       0xbeef     48879
[   0x808] IND_CTXT_DATA_1                                       0          0
```

Figure 28 - Register dump

See the Appendix section for output from 'reg dump' command.

dmactl Utilities

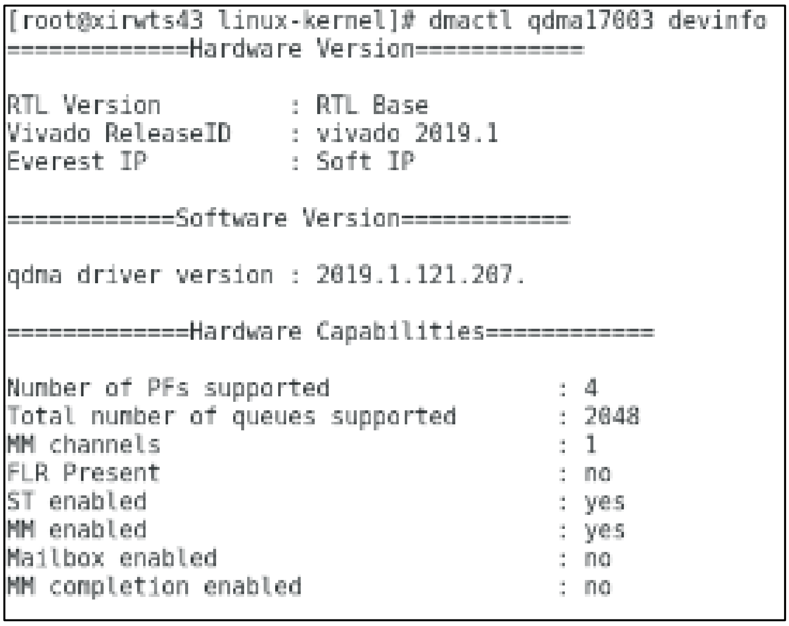
The dmactl tool has several other uses, including providing configuration information on the qdma device, statistics on the number of packets it has sent, and information on descriptor, completion and ring entries.

QDMA device configuration information

The following command can be used to display the configuration information of a QDMA device.

- dmactl qdma<bbddf> devinfo

This command will provide information on the version and capabilities of the hardware that is connected as well as its software version.



```
[root@xirwts43 linux-kernel]# dmactl qdma17003 devinfo
=============Hardware Version============

RTL Version          : RTL Base
Vivado ReleaseID     : vivado 2019.1
Everest IP           : Soft IP

============Software Version============

qdma driver version : 2019.1.121.207.

=============Hardware Capabilities============

Number of PFs supported                  : 4
Total number of queues supported         : 2048
MM channels                              : 1
FLR Present                              : no
ST enabled                               : yes
MM enabled                               : yes
Mailbox enabled                          : no
MM completion enabled                    : no
```

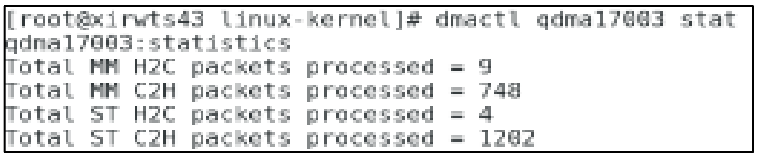
Figure 29 - Displaying a QDMA device's configuration info

QDMA device statistics

The following command can be used to display the statistics of a QDMA device, i.e. the number of packets sent, regardless of the size of the packet.

- dmactl qdma<bbddf> stat

This command displays the h2c and the c2h packets sent for both memory mapped and streaming channels.



```
[root@xirwts43 linux-kernel]# dmactl qdma17003 stat
qdma17003:statistics
Total MM H2C packets processed = 9
Total MM C2H packets processed = 748
Total ST H2C packets processed = 4
Total ST C2H packets processed = 1202
```

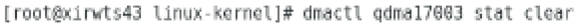
Figure 30 – A qdma device statistics

Clearing QDMA device statistics

The following command can be used to clear the statistics of a QDMA device.

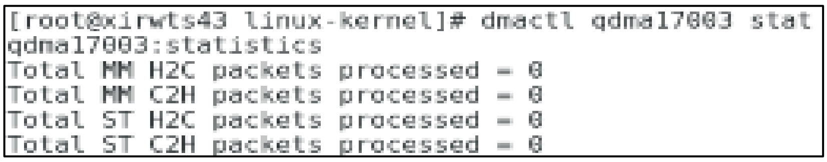
- dmactl qdma<bbddf> stat clear

This command clears the stored information on the number of h2c and c2h packets sent for both memory mapped and streaming channels.



```
[root@xirwts43 linux-kernel]# dmactl qdma17003 stat clear
```

Figure 31: Clearing a qdma devices statistics



```
[root@xirwts43 linux-kernel]# dmactl qdma17003 stat
qdma17003:statistics
Total MM H2C packets processed = 0
Total MM C2H packets processed = 0
Total ST H2C packets processed = 0
Total ST C2H packets processed = 0
```

Figure 32 - A qdma device's statistics after being cleared

Decoding Queue Parameters

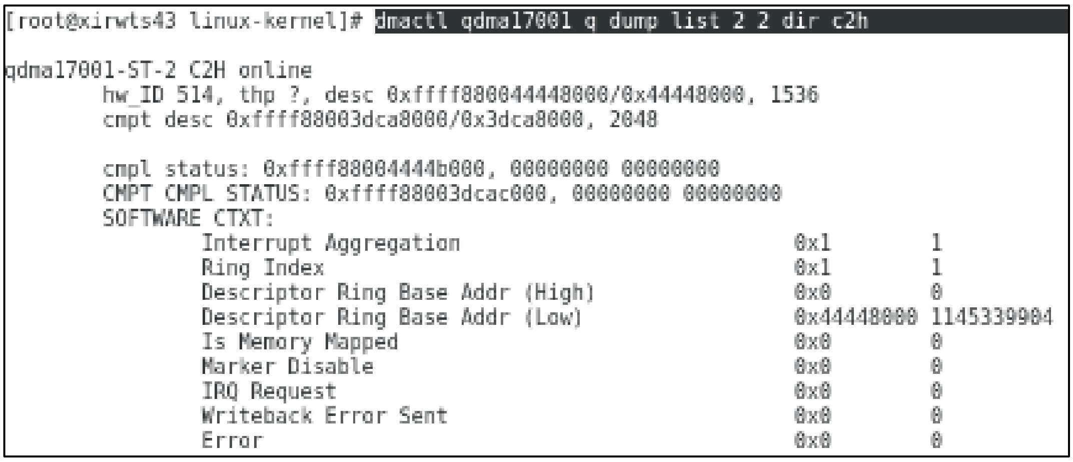
The following command can be used to display a queue's hardware and software parameters.

- dmactl qdma<bbddf> q dump idx <N> dir <h2c|c2h|bi>

This information is decoded.
The following command can also be used to display the parameters for a list of queues at the same time.

- dmactl qdma<bbddf> q dump list <N> <Y> dir <h2c|c2h|bi>

Note: see the Appendix section for the output from queue parameters dump.



```
[root@xirwts43 linux-kernel]# dmactl qdma17001 q dump list 2 2 dir c2h

qdma17001-ST-2 C2H online
        hw_ID 514, thp ?, desc 0xffff880044448000/0x44448000, 1536
        cmpt desc 0xffff88003dca8000/0x3dca8000, 2048

        cmpl status: 0xffff88004444b000, 00000000 00000000
        CMPT CMPL STATUS: 0xffff88003dcac000, 00000000 00000000
        SOFTWARE CTXT:
                Interrupt Aggregation                           0x1        1
                Ring Index                                      0x1        1
                Descriptor Ring Base Addr (High)                0x0        0
                Descriptor Ring Base Addr (Low)                 0x44448000 1145339904
                Is Memory Mapped                                0x0        0
                Marker Disable                                  0x0        0
                IRQ Request                                     0x0        0
                Writeback Error Sent                            0x0        0
                Error                                           0x0        0
```

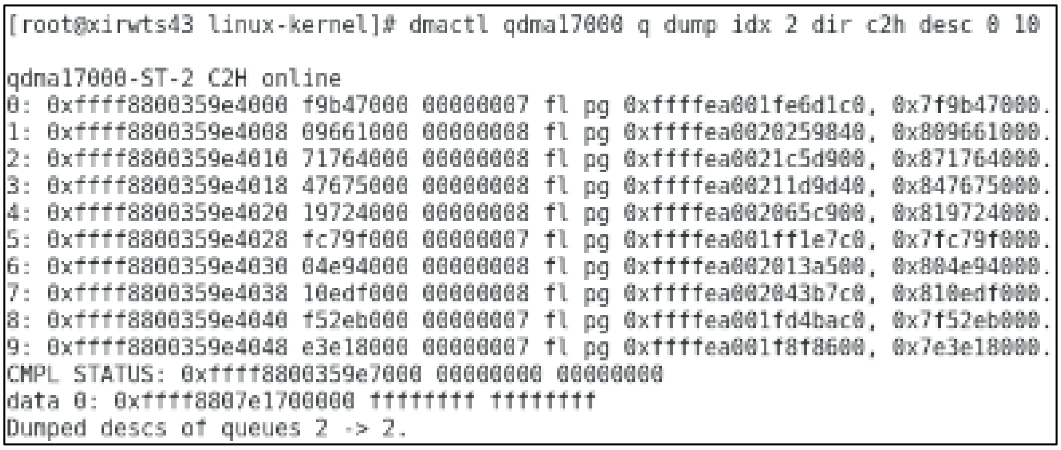
Figure 33 - Displaying a list of queue parameters

Dumping specific descriptors of a queue

The descriptors that are to be dumped can also be specified. This can be done by using the following command

- dmactl qdma<bbddf> q dump idx <N> dir <h2c|c2h|bi> desc <X> <Y>

Where N is the index number of the queue, and X is the start for the list of descriptors and Y is the number of descriptors to be dumped.



```
[root@xirwts43 linux-kernel]# dmactl qdma17000 q dump idx 2 dir c2h desc 0 10

qdma17000-ST-2 C2H online
0: 0xffff8800359e4000 f9b47000 00000007 fl pg 0xffffea001fe6d1c0, 0x7f9b47000.
1: 0xffff8800359e4008 09661000 00000008 fl pg 0xffffea0020259840, 0x809661000.
2: 0xffff8800359e4010 71764000 00000008 fl pg 0xffffea0021c5d900, 0x871764000.
3: 0xffff8800359e4018 47675000 00000008 fl pg 0xffffea00211d9d40, 0x847675000.
4: 0xffff8800359e4020 19724000 00000008 fl pg 0xffffea002065c900, 0x819724000.
5: 0xffff8800359e4028 fc79f000 00000007 fl pg 0xffffea001ff1e7c0, 0x7fc79f000.
6: 0xffff8800359e4030 04e94000 00000008 fl pg 0xffffea002013a500, 0x804e94000.
7: 0xffff8800359e4038 10edf000 00000008 fl pg 0xffffea002043b7c0, 0x810edf000.
8: 0xffff8800359e4040 f52eb000 00000007 fl pg 0xffffea001fd4bac0, 0x7f52eb000.
9: 0xffff8800359e4048 e3e18000 00000007 fl pg 0xffffea001f8f8600, 0x7e3e18000.
CMPL STATUS: 0xffff8800359e7000 00000000 00000000
data 0: 0xffff8807e1700000 ffffffff ffffffff
Dumped descs of queues 2 -> 2.
```

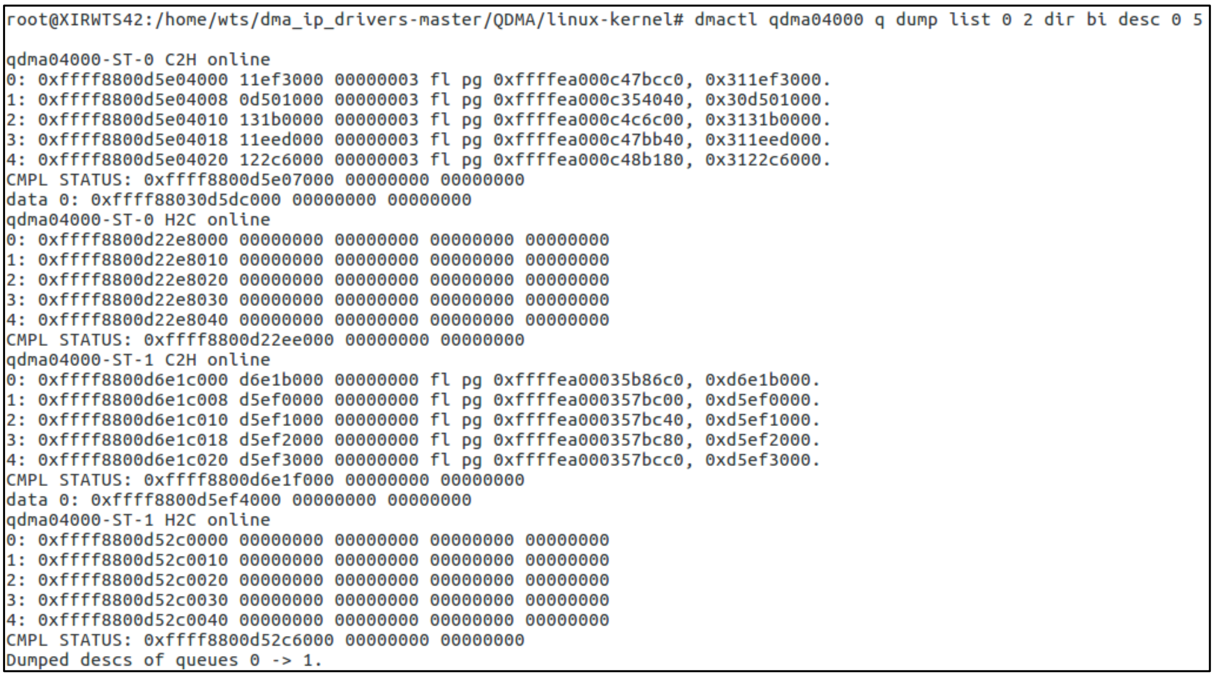
Figure 34 - Dump of specified QDMA descriptors

Dumping specific descriptors for a list of queues

The descriptors that are to be dumped can also be specified. This can be done with the following command.

- dmactl qdma<bbddf> q dump list <N> <I> dir <h2c|c2h|bi> desc <X> <Y>

Where N is the index number of the queue, I is the number of queues, X is the start for the list of descriptors and Y is the number of descriptors to be dumped.

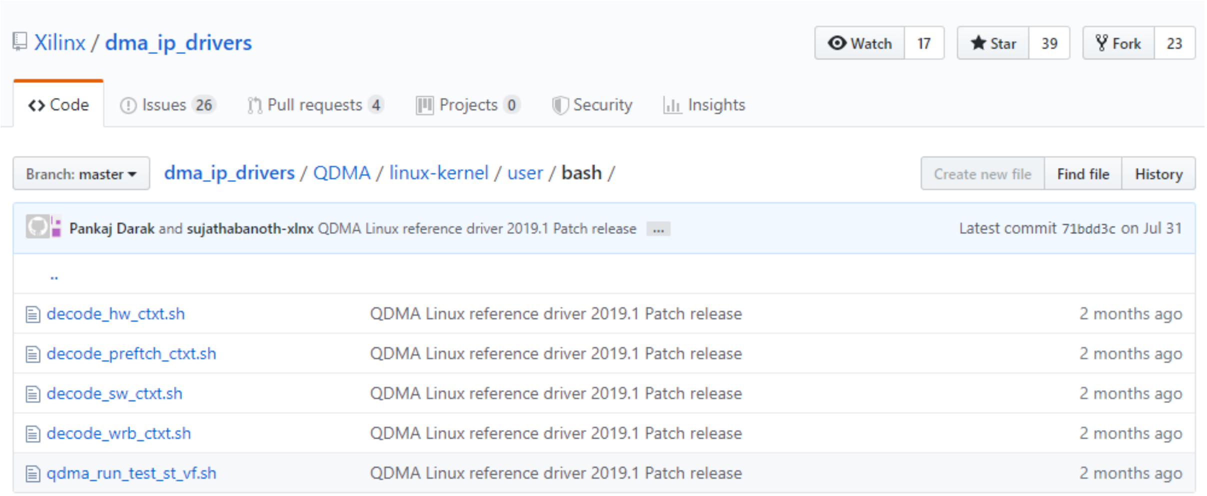


```
root@XIRWTS42:/home/wts/dma_ip_drivers-master/QDMA/linux-kernel# dmactl qdma04000 q dump list 0 2 dir bi desc 0 5

qdma04000-ST-0 C2H online
0: 0xffff8800d5e04000 11ef3000 00000003 fl pg 0xffffea000c47bcc0, 0x311ef3000.
1: 0xffff8800d5e04008 0d501000 00000003 fl pg 0xffffea000c354040, 0x30d501000.
2: 0xffff8800d5e04010 131b0000 00000003 fl pg 0xffffea000c4c6c00, 0x3131b0000.
3: 0xffff8800d5e04018 11eed000 00000003 fl pg 0xffffea000c47bb40, 0x311eed000.
4: 0xffff8800d5e04020 122c6000 00000003 fl pg 0xffffea000c48b180, 0x3122c6000.
CMPL STATUS: 0xffff8800d5e07000 00000000 00000000
data 0: 0xffff88030d5dc000 00000000 00000000
qdma04000-ST-0 H2C online
0: 0xffff8800d22e8000 00000000 00000000 00000000 00000000
1: 0xffff8800d22e8010 00000000 00000000 00000000 00000000
2: 0xffff8800d22e8020 00000000 00000000 00000000 00000000
3: 0xffff8800d22e8030 00000000 00000000 00000000 00000000
4: 0xffff8800d22e8040 00000000 00000000 00000000 00000000
CMPL STATUS: 0xffff8800d22ee000 00000000 00000000
qdma04000-ST-1 C2H online
0: 0xffff8800d6e1c000 d6e1b000 00000000 fl pg 0xffffea00035b86c0, 0xd6e1b000.
1: 0xffff8800d6e1c008 d5ef0000 00000000 fl pg 0xffffea000357bc00, 0xd5ef0000.
2: 0xffff8800d6e1c010 d5ef1000 00000000 fl pg 0xffffea000357bc40, 0xd5ef1000.
3: 0xffff8800d6e1c018 d5ef2000 00000000 fl pg 0xffffea000357bc80, 0xd5ef2000.
4: 0xffff8800d6e1c020 d5ef3000 00000000 fl pg 0xffffea000357bcc0, 0xd5ef3000.
CMPL STATUS: 0xffff8800d6e1f000 00000000 00000000
data 0: 0xffff8800d5ef4000 00000000 00000000
qdma04000-ST-1 H2C online
0: 0xffff8800d52c0000 00000000 00000000 00000000 00000000
1: 0xffff8800d52c0010 00000000 00000000 00000000 00000000
2: 0xffff8800d52c0020 00000000 00000000 00000000 00000000
3: 0xffff8800d52c0030 00000000 00000000 00000000 00000000
4: 0xffff8800d52c0040 00000000 00000000 00000000 00000000
CMPL STATUS: 0xffff8800d52c6000 00000000 00000000
Dumped descs of queues 0 -> 1.
```

Figure 35: Dumping of specified descriptors for a list of queues

Decoding Scripts

The driver example files provide scripts to decode various context values. The provided list of scripts is shown in the figure below.



Xilinx / dma_ip_drivers

dma_ip_drivers / QDMA / linux-kernel / user / bash /

Pankaj Darak and sujathabanoth-xlnx QDMA Linux reference driver 2019.1 Patch release ... Latest commit 71bdd3c on Jul 31

..

decode_hw_ctxt.sh QDMA Linux reference driver 2019.1 Patch release 2 months ago

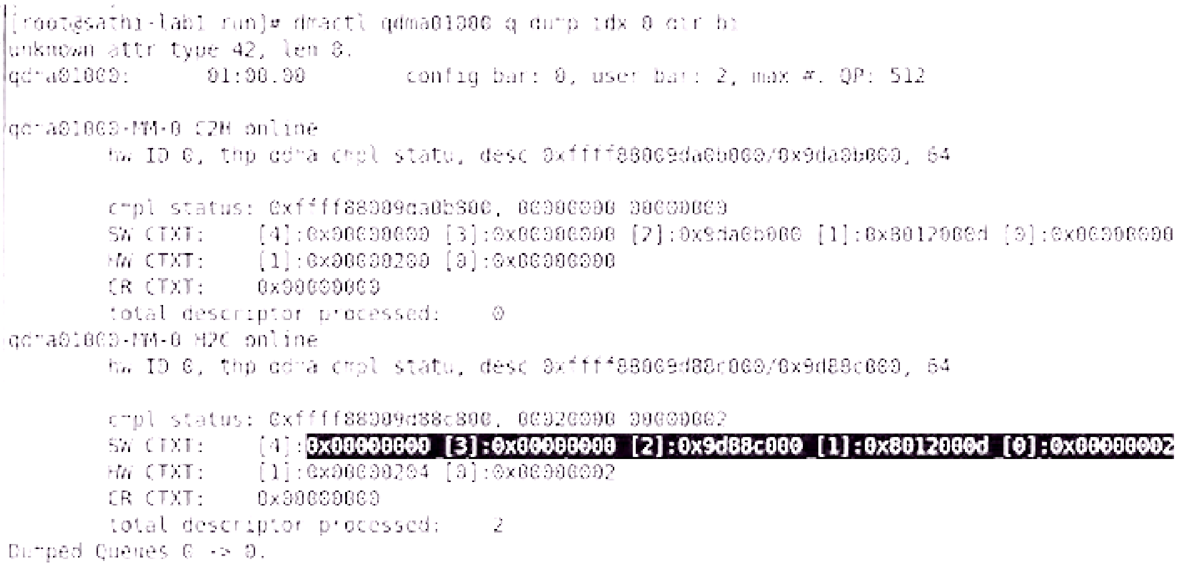
decode_preftch_ctxt.sh QDMA Linux reference driver 2019.1 Patch release 2 months ago

decode_sw_ctxt.sh QDMA Linux reference driver 2019.1 Patch release 2 months ago

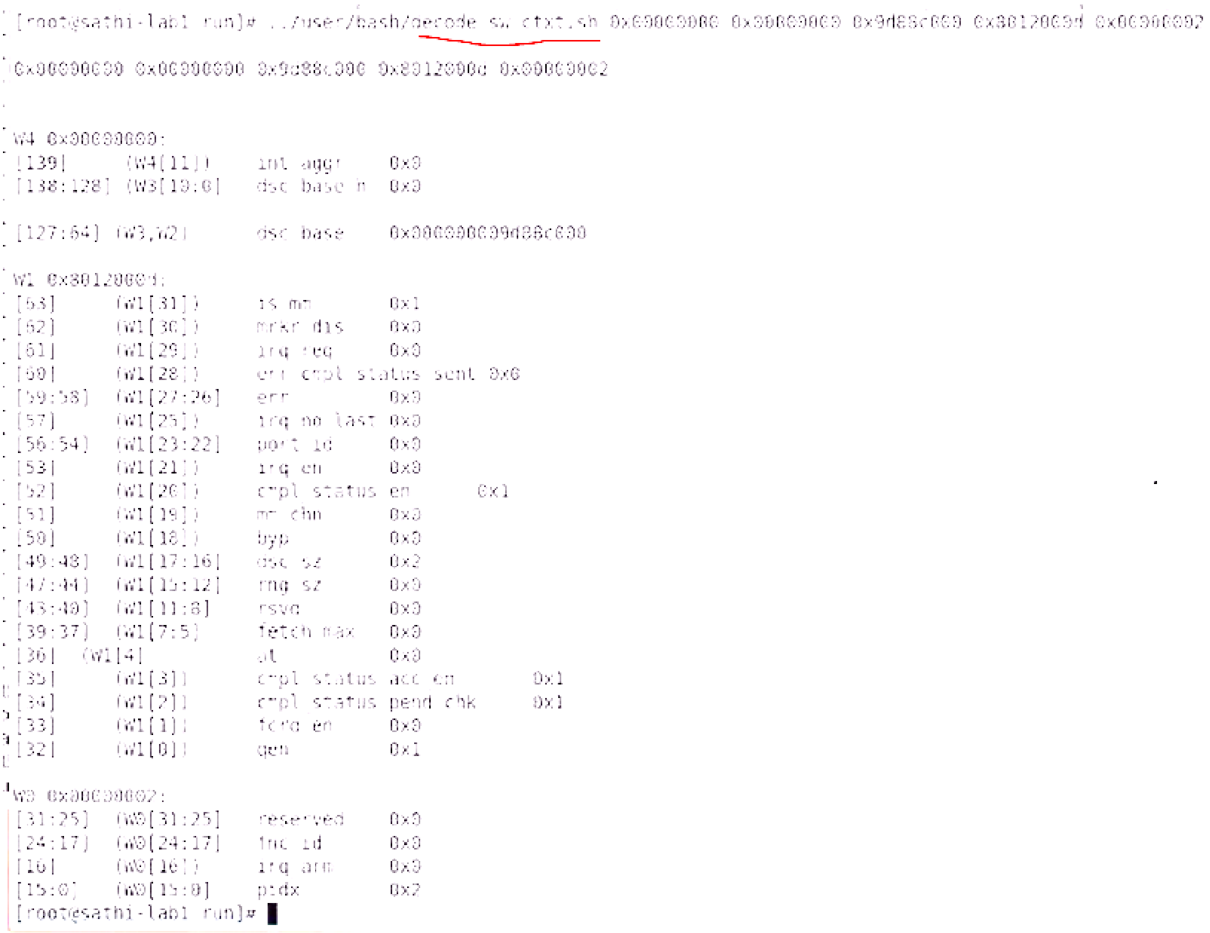
decode_wrb_ctxt.sh QDMA Linux reference driver 2019.1 Patch release 2 months ago

qdma_run_test_st_vf.sh QDMA Linux reference driver 2019.1 Patch release 2 months ago

The script can be used as shown below:



```
[root@sathi-lab1 run]# dmactl qdma01000 q dump idx 0 dir h2c
unknown attr type 42, len 8.
qdma01000:      01:00.00        config bar: 0, user bar: 2, max #. QP: 512

qdma01000-MM-0 C2H online
        hw ID 0, thp qdma cmpl statu, desc 0xffff88009da0b000/0x9da0b000, 64

        cmpl status: 0xffff88009da0b800, 00000000 00000000
        SW CTXT:    [4]:0x00000000 [3]:0x00000000 [2]:0x9da0b000 [1]:0x8012000d [0]:0x00000000
        HW CTXT:    [1]:0x00000200 [0]:0x00000000
        CR CTXT:    0x00000000
        total descriptor processed:    0
qdma01000-MM-0 H2C online
        hw ID 0, thp qdma cmpl statu, desc 0xffff88009d88c000/0x9d88c000, 64

        cmpl status: 0xffff88009d88c800, 00020000 00000002
        SW CTXT:    [4]:0x00000000 [3]:0x00000000 [2]:0x9d88c000 [1]:0x8012000d [0]:0x00000002
        HW CTXT:    [1]:0x00000204 [0]:0x00000002
        CR CTXT:    0x00000000
        total descriptor processed:    2
Dumped Queues 0 -> 0.
```

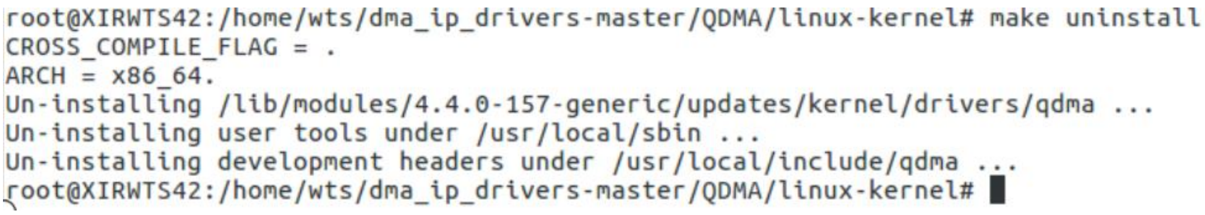


```
[root@sathi-lab1 run]# ../user/bash/decode_sw_ctxt.sh 0x00000000 0x00000000 0x9d88c000 0x8012000d 0x00000002

0x00000000 0x00000000 0x9d88c000 0x8012000d 0x00000002


W4 0x00000000:
[139]       (W4[11])    int aggr     0x0
[138:128] (W3[10:0]    dsc base h   0x0

[127:64] (W3,W2)       dsc base     0x000000009d88c000

W1 0x8012000d:
[63]       (W1[31])     is mm        0x1
[62]       (W1[30])     mrkr dis     0x0
[61]       (W1[29])     irq req      0x0
[60]       (W1[28])     err cmpl status sent 0x0
[59:58]   (W1[27:26]   err          0x0
[57]       (W1[25])     irq no last  0x0
[56:54]   (W1[23:22]   port id      0x0
[53]       (W1[21])     irq en       0x0
[52]       (W1[20])     cmpl status en        0x1
[51]       (W1[19])     mm chn       0x0
[50]       (W1[18])     byp          0x0
[49:48]   (W1[17:16]   dsc sz       0x2
[47:44]   (W1[15:12]   rng sz       0x0
[43:40]   (W1[11:8]    rsvd         0x0
[39:37]   (W1[7:5]     fetch max    0x0
[36]   (W1[4]          at           0x0
[35]       (W1[3])      cmpl status acc en        0x1
[34]       (W1[2])      cmpl status pend chk      0x1
[33]       (W1[1])      fcrd en      0x0
[32]       (W1[0])      qen          0x1

W0 0x00000002:
[31:25]   (W0[31:25]   reserved     0x0
[24:17]   (W0[24:17]   fnc id       0x0
[16]       (W0[16])     irq arm      0x0
[15:0]    (W0[15:0]    pidx         0x2
[root@sathi-lab1 run]# 
```

Removing the Driver and the Modules

If the Linux QDMA driver software and the modules associated with it have been installed before, and it is to be reinstalled or removed, run the following commands.

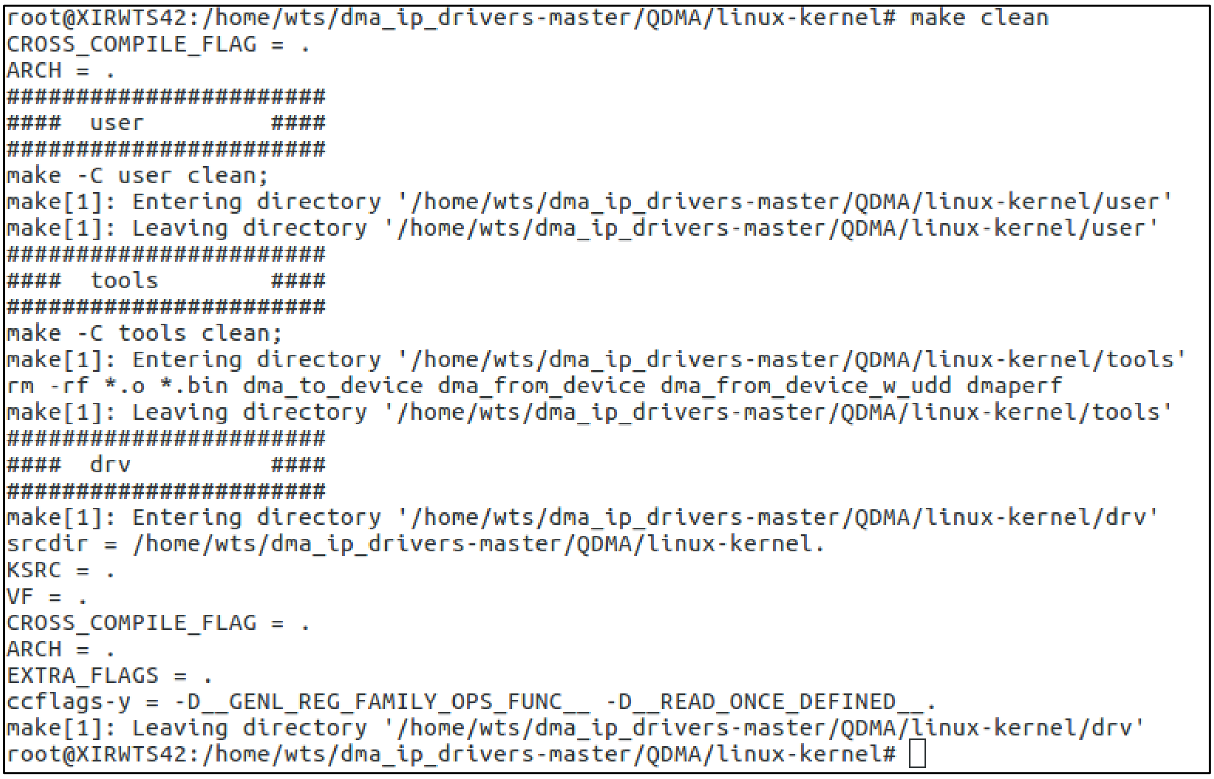
- make uninstall



```
root@XIRWTS42:/home/wts/dma_ip_drivers-master/QDMA/linux-kernel# make uninstall
CROSS_COMPILE_FLAG = .
ARCH = x86_64.
Un-installing /lib/modules/4.4.0-157-generic/updates/kernel/drivers/qdma ...
Un-installing user tools under /usr/local/sbin ...
Un-installing development headers under /usr/local/include/qdma ...
root@XIRWTS42:/home/wts/dma_ip_drivers-master/QDMA/linux-kernel# 
```

Figure 36 - Uninstalling the driver

- make clean

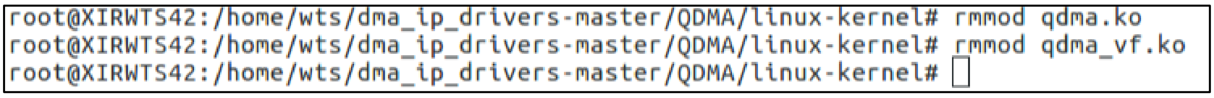


```
root@XIRWTS42:/home/wts/dma_ip_drivers-master/QDMA/linux-kernel# make clean
CROSS_COMPILE_FLAG = .
ARCH = .
#######################
####  user         ####
#######################
make -C user clean;
make[1]: Entering directory '/home/wts/dma_ip_drivers-master/QDMA/linux-kernel/user'
make[1]: Leaving directory '/home/wts/dma_ip_drivers-master/QDMA/linux-kernel/user'
#######################
####  tools        ####
#######################
make -C tools clean;
make[1]: Entering directory '/home/wts/dma_ip_drivers-master/QDMA/linux-kernel/tools'
rm -rf *.o *.bin dma_to_device dma_from_device dma_from_device_w_udd dmaperf
make[1]: Leaving directory '/home/wts/dma_ip_drivers-master/QDMA/linux-kernel/tools'
#######################
####  drv          ####
#######################
make[1]: Entering directory '/home/wts/dma_ip_drivers-master/QDMA/linux-kernel/drv'
srcdir = /home/wts/dma_ip_drivers-master/QDMA/linux-kernel.
KSRC = .
VF = .
CROSS_COMPILE_FLAG = .
ARCH = .
EXTRA_FLAGS = .
ccflags-y = -D__GENL_REG_FAMILY_OPS_FUNC__ -D__READ_ONCE_DEFINED__.
make[1]: Leaving directory '/home/wts/dma_ip_drivers-master/QDMA/linux-kernel/drv'
root@XIRWTS42:/home/wts/dma_ip_drivers-master/QDMA/linux-kernel# 
```

Figure 37 - Cleaning the linux-kernal directory

Use the following command to remove the qdma modules before reinstalling them:

- rmmod qdma.ko
- rmmod qdma_vf.ko



```
root@XIRWTS42:/home/wts/dma_ip_drivers-master/QDMA/linux-kernel# rmmod qdma.ko
root@XIRWTS42:/home/wts/dma_ip_drivers-master/QDMA/linux-kernel# rmmod qdma_vf.ko
root@XIRWTS42:/home/wts/dma_ip_drivers-master/QDMA/linux-kernel# 
```

Figure 38 - Removing qdma modules

Appendix

Output of ‘make’ command

```
"distro=RHEL, dmajor=7 dminor=4 "
3.10.0-693.el7.x86_64: RHEL,7,4, -I/lib/modules/3.10.0-693.el7.x86_64/source/arch/x86/include/generated -DRHEL7SP4 -DRHEL7
CROSS_COMPILE_FLAG = .
ARCH = x86_64.
#######################
####  user        ####
#######################
make -C user
make[1]: Entering directory `/root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/user'
cc -g -I. -I../include    -c -o cli/reg_cmd.o cli/reg_cmd.c
cc -g -I. -I../include    -c -o cli/cmd_parse.o cli/cmd_parse.c
cc -g -I. -I../include    -c -o cli/nl_user.o cli/nl_user.c
cc -g -I. -I../include    -c -o cli/main.o cli/main.c
cc cli/reg_cmd.o cli/cmd_parse.o cli/nl_user.o cli/main.o -o dmactl
make[1]: Leaving directory `/root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/user'
#######################
####  tools        ####
#######################
make -C tools
make[1]: Entering directory `/root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/tools'
cc   -g   -c   -std=c99   -o   dma_to_device.o   dma_to_device.c   -D_FILE_OFFSET_BITS=64   -D_GNU_SOURCE   -D_LARGE_FILE_SOURCE   -
D_AIO_AIX_SOURCE
cc -lrt -o dma_to_device dma_to_device.o -D_FILE_OFFSET_BITS=64 -D_GNU_SOURCE -D_LARGE_FILE_SOURCE
cc   -g   -c   -std=c99   -o   dma_from_device.o   dma_from_device.c   -D_FILE_OFFSET_BITS=64   -D_GNU_SOURCE   -D_LARGE_FILE_SOURCE   -
D_AIO_AIX_SOURCE
cc -lrt -o dma_from_device dma_from_device.o -D_FILE_OFFSET_BITS=64 -D_GNU_SOURCE -D_LARGE_FILE_SOURCE
cc -g -c -std=c99 -o dma_from_device_w_udd.o dma_from_device_w_udd.c -D_FILE_OFFSET_BITS=64 -D_GNU_SOURCE -D_LARGE_FILE_SOURCE
-D_AIO_AIX_SOURCE
cc -lrt -o dma_from_device_w_udd dma_from_device_w_udd.o -D_FILE_OFFSET_BITS=64 -D_GNU_SOURCE -D_LARGE_FILE_SOURCE
cc -g -c -std=c99 -o dmaperf.o dmaperf.c -D_FILE_OFFSET_BITS=64 -D_GNU_SOURCE -D_LARGE_FILE_SOURCE -D_AIO_AIX_SOURCE
cc -pthread -lrt -o dmaperf dmaperf.o -laio -D_FILE_OFFSET_BITS=64 -D_GNU_SOURCE -D_LARGE_FILE_SOURCE
make[1]: Leaving directory `/root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/tools'
#######################
#### PF drv          ####
#######################
srcdir = /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel.
KSRC = /lib/modules/3.10.0-693.el7.x86_64/source.
VF = 0.
CROSS_COMPILE_FLAG = .
ARCH = x86_64.
EXTRA_FLAGS = .
ccflags-y = -D__READ_ONCE_DEFINED__.
make[1]: Entering directory `/root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv'
make[2]: Entering directory `/usr/src/kernels/3.10.0-693.el7.x86_64'
srcdir = /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel.
KSRC = /lib/modules/3.10.0-693.el7.x86_64/source.
VF = 0.
CROSS_COMPILE_FLAG = .
ARCH = x86_64.
EXTRA_FLAGS = .
ccflags-y =  -D__READ_ONCE_DEFINED__.
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/nl.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/cdev.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/qdma_mod.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_mbox.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_intr.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_st_c2h.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_thread.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/libqdma_export.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_context.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_sriov.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_platform.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_descq.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_regs.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_debugfs_queue.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_debugfs.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_debugfs_dev.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/libqdma_config.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_device.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/xdev.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/thread.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_access/qdma_mbox_protocol.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_access/qdma_access.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_access/qdma_list.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_access/qdma_resource_mgmt.o
  LD [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/qdma.o
  Building modules, stage 2.
srcdir = /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel.
KSRC = /lib/modules/3.10.0-693.el7.x86_64/source.
VF = 0.
CROSS_COMPILE_FLAG = .
ARCH = x86_64.
EXTRA_FLAGS = .
ccflags-y = -D__READ_ONCE_DEFINED__.
  MODPOST 1 modules
  CC      /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/qdma.mod.o
  LD [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/qdma.ko
make[2]: Leaving directory `/usr/src/kernels/3.10.0-693.el7.x86_64'
make[1]: Leaving directory `/root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv'
#######################
```

```
#### VF drv          ####
#######################
srcdir = /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel.
KSRC = /lib/modules/3.10.0-693.el7.x86_64/source.
VF = 1.
CROSS_COMPILE_FLAG = .
ARCH = x86_64.
EXTRA_FLAGS = -D__QDMA_VF__.
ccflags-y = -D__READ_ONCE_DEFINED__.
make[1]: Entering directory `/root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv'
make[2]: Entering directory `/usr/src/kernels/3.10.0-693.el7.x86_64'
srcdir = /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel.
KSRC = /lib/modules/3.10.0-693.el7.x86_64/source.
VF = 1.
CROSS_COMPILE_FLAG = .
ARCH = x86_64.
EXTRA_FLAGS = -D__QDMA_VF__.
ccflags-y =  -D__READ_ONCE_DEFINED__.
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/nl.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/cdev.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/qdma_mod.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/qdma.mod.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_debugfs_queue.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_mbox.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_st_c2h.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_thread.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/libqdma_export.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_intr.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_debugfs.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_context.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_sriov.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_platform.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_descq.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_regs.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_debugfs_dev.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/libqdma_config.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_device.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/xdev.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/thread.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_access/qdma_mbox_protocol.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_access/qdma_access.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_access/qdma_list.o
  CC [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/libqdma/qdma_access/qdma_resource_mgmt.o
  LD [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/qdma_vf.o
  Building modules, stage 2.
srcdir = /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel.
KSRC = /lib/modules/3.10.0-693.el7.x86_64/source.
VF = 1.
CROSS_COMPILE_FLAG = .
ARCH = x86_64.
EXTRA_FLAGS = -D__QDMA_VF__.
ccflags-y = -D__READ_ONCE_DEFINED__.
  MODPOST 1 modules
  CC      /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/qdma_vf.mod.o
  LD [M]  /root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv/qdma_vf.ko
make[2]: Leaving directory `/usr/src/kernels/3.10.0-693.el7.x86_64'
make[1]: Leaving directory `/root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/drv'
```

Output of 'make install' command

```
"distro=RHEL, dmajor=7 dminor=4 "
3.10.0-693.el7.x86_64: RHEL,7,4, -I/lib/modules/3.10.0-693.el7.x86_64/source/arch/x86/include/generated -DRHEL7SP4 -DRHEL7
CROSS_COMPILE_FLAG = .
ARCH = x86_64.
installing kernel modules to /lib/modules/3.10.0-693.el7.x86_64/updates/kernel/drivers/qdma ...
'/root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/build/qdma.ko'          ->          '/lib/modules/3.10.0-
693.el7.x86_64/updates/kernel/drivers/qdma/qdma.ko'
'/root/Downloads/QDMA/dma_ip_drivers-master/QDMA/linux-kernel/build/qdma_vf.ko'         ->         '/lib/modules/3.10.0-
693.el7.x86_64/updates/kernel/drivers/qdma/qdma_vf.ko'
installing user tools to /usr/local/sbin ...
'build/dmactl' -> '/usr/local/sbin/dmactl'
'tools/dma_from_device' -> '/usr/local/sbin/dma_from_device'
'tools/dma_to_device' -> '/usr/local/sbin/dma_to_device'
'tools/dmaperf' -> '/usr/local/sbin/dmaperf'
MAN PAGES:
removed '/usr/share/man/man8/dmactl.8.gz'
'docs/dmactl.8.gz' -> '/usr/share/man/man8/dmactl.8.gz'
install Xilinx DMA udev rules:
removed '/etc/udev/rules.d/30-qdma.rules'
'etc/30-qdma.rules' -> '/etc/udev/rules.d/30-qdma.rules'
installing development headers to /usr/local/include/qdma ...
'include/qdma_nl.h' -> '/usr/local/include/qdma/qdma_nl.h'
'include/qdma_reg_dump.h' -> '/usr/local/include/qdma/qdma_reg_dump.h'
'include/qdma_user_reg_dump.h' -> '/usr/local/include/qdma/qdma_user_reg_dump.h'
'include/xdev_regs.h' -> '/usr/local/include/qdma/xdev_regs.h'
```

Output of 'man dmactl'

```
dmactl(8)                                                                                  System Manager's Manual
dmactl(8)
```

```
NAME
       dmactl - QDMA configuration and control utility


SYNOPSIS
       dmactl [dev | qdma[vf]<bbddf>] [operation]
       dmactl [--help] | [-h]

DESCRIPTION
       dmactl is a QDMA control utility which allows administration of the Xilinx QDMA queues.

       It can perform the following functions:

           - Query the QDMA functions/devices the driver has bind into
           - Query control and configuration
                  > List all of the queues on a device/function
                  > Add/configure a new queues on a device/function
                  > Start an already added/configured queue (i.e., bring the queue online)
                  > Stop an started queue (i.e., bring the queue offline)
                  > Delete an already added/configured queue
           - register access
                  > Read a register
                  > Write a register
                  > Dump the qdma config bar and user bar registers
           - debug helper
                  > Display a queue's configuration parameters
                  > Display a queue's descriptor ring entries
                  > Display a c2h queue's completion ring entries
                  > Display the interrupt ring entries
           - For help run
                  > dmactl -h

OPTIONS
   Device Info commands
       dev    [operation] - System wide device operations
              list - List all QDMA functions

   Version Command
       qdma<bbddf> devinfo  - lists the Hardware and Software version and capabilities

Device Stat Command
qdma<bbddf> operation - Device statistic command
              stat         - statistics accumulated for this device
              stat clear - clear the statistics accumulated for this device

Queue commands
qdma<bbddf> [operation]  - Per QDMA device operation
Queue list - List available queues in a PF/VF
q list - list available queues
Example: dmactl  qdma<bbddf> q list /* To list available queues in PF/VF '<bbddf>' */

Queue Add - Add a queue to a PF/VF
q add idx <N> [mode <st|mm>] [dir <h2c|c2h|bi>] - add a queue
mode default to mm
dir default to h2c
Example: dmactl  qdma<bbddf> q add idx Y  mode st  dir c2h /* To add q of index 'Y' to PF/VF '<bbddf>' */

q add list <start_idx> <N>  [ mode <st|mm> ] [ dir <h2c|c2h|bi> ] - add list of queues

mode default to mm

dir default to h2c
Example: dmactl  qdma<bbddf> q add list 0 10  mode st  dir c2h /* To add 10 queues from index 0 to PF/VF '<bbddf>' */

                         st   - Streaming mode
                         mm   - Memory mapped mode
                         h2c  - Host to card
                         c2h  - Card to host
                         bi   - both h2c and c2h

 Queue Start  - Start an already added queue
q start idx <N> [dir <h2c|c2h|bi>] [en_mm_cmpl] [idx_ringsz <0:15>] [idx_bufsz <0:15>] [idx_tmr <0:15>] [idx_cntr <0:15>]
[trigmode <every|usr_cnt|usr|usr_tmr|dis>] [cmptsz <0|1|2|3>]
[sw_desc_sz    <3>]    [desc_bypass_en]    [pfetch_en]    [pfetch_bypass_en]    [dis_cmpl_status]    [dis_cmpl_status_acc]
[dis_cmpl_status_pend_chk] [c2h_udd_en] [dis_fetch_credit] [dis_cmpt_stat]
[c2h_cmpl_intr_en] [cmpl_ovf_dis] - start a queue
Example: dmactl  qdma<bbddf> q start idx Y  dir c2h /* To start q of index 'Y' of PF/VF '<bbddf>' */

q start list <start_idx> <N> [dir <h2c|c2h|bi>] [en_mm_cmpl] [idx_ringsz <0:15>] [idx_bufsz <0:15>] [idx_tmr <0:15>] [idx_cntr
<0:15>] [trigmode <every|usr_cnt|usr|usr_tmr|dis>]
[cmptsz <0|1|2|3>] [sw_desc_sz <3>] [desc_bypass_en] [pfetch_en] [pfetch_bypass_en] [dis_cmpl_status] [dis_cmpl_status_acc]
[dis_cmpl_status_pend_chk] [c2h_udd_en] [dis_fetch_credit]
[dis_cmpt_stat] [c2h_cmpl_intr_en] [cmpl_ovf_dis]
Example: dmactl  qdma<bbddf> q start list 0 10  dir c2h /* To start 10 queues from 0 of PF/VF '<bbddf>' */
Note:  The default values of all attributes will be 0 if not provided.

-   Default value of dir  will be 'h2c'.
-   cmptsz , trigmode , idx_cntr , idx_tmr , c2h_cmpl_intr_en , cmpl_ovf_dis  - streaming c2h only options

Queue Stop  - Stop an already started queue
q stop idx <N> [dir <h2c|c2h|bi>] - stop a queue
Example: dmactl  qdma<bbddf> q stop idx Y  dir c2h /* To stop q of index 'Y' of PF/VF '<bbddf>' */

q stop list <start_idx> <N> [dir <h2c|c2h|bi>] - stop list of queues
Example: dmactl  qdma<bbddf> q stop list 0 10  dir c2h /* To stop 10 queues from index 0 of PF/VF '<bbddf>' */

Queue Delete  - Delete queue from a PF/VF
q del idx <N> [dir <h2c|c2h|bi>] - delete a queue
```

```
Example: dmactl  qdma<bbddf> q del idx Y  dir c2h /* To delete q of index 'Y' of PF/VF '<bbddf>' */

q del list <start_idx> <N> [dir <h2c|c2h|bi>] - delete list of queues
Example: dmactl  qdma<bbddf> q del list 0 10  dir c2h /* To delete 10 queues from index 0 of PF/VF '<bbddf>' */

Queue Dump

q dump idx <N> [dir <h2c|c2h|bi>] - dump queue parameters
Example: dmactl  qdma<bbddf> q dump idx Y  dir c2h /* To dump q of index 'Y' of PF/VF '<bbddf>' */

q dump list <start_idx> <N> [dir <h2c|c2h|bi> ] - dump list of queue parameters
Example: dmactl  qdma<bbddf> q dump list 0 10  dir c2h /* To dump 10 queues from index 0 of PF/VF '<bbddf>' */

Queue Descriptor Dump
q dump idx <N> [dir <h2c|c2h|bi>] [desc <x> <y>] - dump queue descriptors
                                <x> -  range start
                                <y> -  range end
Example: dmactl  qdma<bbddf> q dump idx Y  dir c2h  desc 0 10 /* To dump descriptors of q index 'Y' of PF/VF '<bbddf>' from 0
to 10 */

q dump list <start_idx> <N> [dir <h2c|c2h|bi>] [desc <x> <y>] - dump descriptors of list of queue

Example: dmactl  qdma<bbddf> q dump list 0 10  dir c2h  desc 0 10 /* To dump descriptors of 10 queues from index 0 of PF/VF
'<bbddf>' from 0 to 10 */

Completion Queue Descriptor Dump
q dump idx <N> [dir <h2c|c2h|bi>] [cmpt <x> <y>] - dump completion entries
Example: dmactl  qdma<bbddf> q dump idx Y  dir c2h  cmpt 0 10 /* To dump completion entries of q index 'Y' of PF/VF '<bbddf>'
from 0 to 10 */

q dump list <start_idx> <N> [dir <h2c|c2h|bi>] [cmpt <x> <y>] - dump completion entries of list of queue
Example: dmactl  qdma<bbddf> q dump list 0 10  dir c2h  cmpt 0 10 /* To dump completion entries of 10 queues from index 0 of
PF/VF '<bbddf>' from 0 to 10 */

Register commands
qdma<bbddf> [operation] - Register operation on this device
reg dump [dmap <Q> <N>] -  register dump. Only dump dmap registers if dmap is specified. Specify dmap range to dump: Q=queue,
N=num of queues.
Example: dmactl  qdma<bbddf> reg dump /* to dump registers of PF/VF <bbddf>' */

reg read bar <N> <addr> -  read a register
Example: dmactl  qdma<bbddf> reg read bar 1 Y /* to read a register of PF/VF '<bbddf>', bar 1, offset Y */

reg write bar <N> <addr> <val> -  write to a register
Example: dmactl  qdma<bbddf> reg write bar 1 Y Z /* to write value 'Z' to a register of PF/VF '<bbddf>', bar 1, offset Y */

Interrupt Ring Dump command
qdma<bbddf> [operation] - Interrupt ring dump for this device
intring dump vector <N> <start_idx> <end_idx>  -  interrupt ring dump for vector number <N>

For interrupt entries :<start_idx> --- <end_idx>
Example: dmactl  qdma<bbddf> intring dump vector N idx<1> idx<n> /* to dump vector 'N' of PF/VF '<bbddf>' from idx<1> to idx<n>
*/

dmactl 2019.1                                                                                                        dmactl(8)
```

Output of ‘dmactl -h’

```
Usage: dmactl [dev|qdma[vf]<N>] [operation]
          dev [operation]: system wide FPGA operations
                    list                        list all qdma functions
          qdma[N] [operation]: per QDMA FPGA operations
                    devinfo                     lists the Hardware and Software version and capabilities
                    stat                        statistics of qdma[N] device
                    stat clear                  clear all statistics data of qdma[N} device
                    q list                      list all queues
                    q add idx <N> [mode <mm|st>] [dir <h2c|c2h|bi>] - add a queue
                                                                      *mode default to mm
                                                                      *dir default to h2c
                    q add list <start_idx> <num_Qs> [mode <mm|st>] [dir <h2c|c2h|bi>] - add multiple queues at once
                    q start idx <N> [dir <h2c|c2h|bi>] [en_mm_cmpl] [idx_ringsz <0:15>] [idx_bufsz <0:15>] [idx_tmr <0:15>]
                                    [idx_cntr   <0:15>]   [trigmode   <every|usr_cnt|usr|usr_tmr|dis>]   [cmptsz   <0|1|2|3>]
[sw_desc_sz <3>]
                                    [desc_bypass_en] [pfetch_en] [pfetch_bypass_en] [dis_cmpl_status]
                                    [dis_cmpl_status_acc] [dis_cmpl_status_pend_chk] [c2h_udd_en]
                                    [cmpl_ovf_dis]  [dis_fetch_credit]  [dis_cmpl_status]  [c2h_cmpl_intr_en]  -  start  a  single
queue
                    q start list <start_idx> <num_Qs> [en_mm_cmpl] [dir <h2c|c2h|bi>] [idx_bufsz <0:15>] [idx_tmr <0:15>]
                                    [idx_cntr   <0:15>]   [trigmode   <every|usr_cnt|usr|usr_tmr|dis>]   [cmptsz   <0|1|2|3>]
[sw_desc_sz <3>]
                                    [desc_bypass_en] [pfetch_en] [pfetch_bypass_en] [dis_cmpl_status]
                                    [dis_cmpl_status_acc] [dis_cmpl_status_pend_chk] [cmpl_ovf_dis]
                                    [dis_fetch_credit] [dis_cmpl_status] [c2h_cmpl_intr_en] - start multiple queues at once
                    q stop idx <N> dir [<h2c|c2h|bi>] - stop a single queue
                    q stop list <start_idx> <num_Qs> dir [<h2c|c2h|bi>] - stop list of queues at once
                    q del idx <N> dir [<h2c|c2h|bi>] - delete a queue
                    q del list <start_idx> <num_Qs> dir [<h2c|c2h|bi>] - delete list of queues at once
                    q dump idx <N> dir [<h2c|c2h|bi>]   dump queue param
                    q dump list <start_idx> <num_Qs> dir [<h2c|c2h|bi>] - dump queue param
                    q dump idx <N> dir [<h2c|c2h|bi>] desc <x> <y> - dump desc ring entry x ~ y
                    q dump list <start_idx> <num_Qs> dir [<h2c|c2h|bi>] desc <x> <y> - dump desc ring entry x ~ y
                    q dump idx <N> dir [<h2c|c2h|bi>] cmpt <x> <y> - dump cmpt ring entry x ~ y
                    q dump list <start_idx> <num_Qs> dir [<h2c|c2h|bi>] cmpt <x> <y> - dump cmpt ring entry x ~ y
                    q cmpt_read idx <N> dir [<h2c|c2h|bi>] - read the completion data
                    reg dump [dmap <Q> <N>]          - register dump. Only dump dmap registers if dmap is specified.
```

```
                                                       specify dmap range to dump: Q=queue, N=num of queues
                    reg read [bar <N>] <addr>         - read a register
                    reg write [bar <N>] <addr> <val> - write a register
                    intring dump vector <N> <start_idx> <end_idx> - interrupt ring dump for vector number <N>
                                                                    for interrupt entries :<start_idx> --- <end_idx>
```

Output of ‘dma_from_device -h’

```
dma_from_device

usage: dma_from_device [OPTIONS]

Read via SGDMA, optionally save output to a file

  -d (--device) device (defaults to /dev/qdma01000-MM-0)
  -a (--address) the start address on the AXI bus
  -s (--size) size of a single transfer in bytes, default 32.
  -o (--offset) page offset of transfer
  -c (--count) number of transfers, default is 1.
  -f (--file) file to write the data of the transfers
  -h (--help) print usage help and exit
  -v (--verbose) verbose output
```

Output of ‘dma_to_device -h’

```
dma_from_device

usage: dma_from_device [OPTIONS]

Read via SGDMA, optionally save output to a file

  -d (--device) device (defaults to /dev/qdma01000-MM-0)
  -a (--address) the start address on the AXI bus
  -s (--size) size of a single transfer in bytes, default 32.
  -o (--offset) page offset of transfer
  -c (--count) number of transfers, default is 1.
  -f (--file) file to write the data of the transfers
  -h (--help) print usage help and exit
  -v (--verbose) verbose output
```

Output for reg dump

```
################################################################################
###                 qdma17000, pci 17:00.00, reg dump
################################################################################

USER BAR #2
[      0] ST_C2H_QID                                      0x1        1
[    0x4] ST_C2H_PKTLEN                                   0x80       128
[    0x8] ST_C2H_CONTROL                                  0          0
[    0xc] ST_H2C_CONTROL                                  0          0
[   0x10] ST_H2C_STATUS                                   0x10       16
[   0x14] ST_H2C_XFER_CNT                                 0x10       16
[   0x20] ST_C2H_PKT_CNT                                  0x1        1
[   0x30] ST_C2H_CMPT_DATA_0                              0          0
[   0x34] ST_C2H_CMPT_DATA_1                              0          0
[   0x38] ST_C2H_CMPT_DATA_2                              0          0
[   0x3c] ST_C2H_CMPT_DATA_3                              0          0
[   0x40] ST_C2H_CMPT_DATA_4                              0          0
[   0x44] ST_C2H_CMPT_DATA_5                              0          0
[   0x48] ST_C2H_CMPT_DATA_6                              0          0
[   0x4c] ST_C2H_CMPT_DATA_7                              0          0
[   0x50] ST_C2H_CMPT_SIZE                                0          0
[   0x60] ST_SCRATCH_REG_0                                0          0
[   0x64] ST_SCRATCH_REG_1                                0          0
[   0x88] ST_C2H_PKT_DROP                                 0          0
[   0x8c] ST_C2H_PKT_ACCEPT                               0          0
[   0x90] DSC_BYPASS_LOOP                                 0          0
[   0x94] USER_INTERRUPT                                  0          0
[   0x98] USER_INTERRUPT_MASK                             0          0
[   0x9c] USER_INTERRUPT_VEC                              0          0
[   0xa0] DMA_CONTROL                                     0          0
[   0xa4] VDM_MSG_READ                                    0          0

CONFIG BAR #0
[      0] CFG_BLOCK_ID                                    0x1fd30000 533921792
[    0x4] CFG_BUSDEV                                      0          0
[    0x8] CFG_PCIE_MAX_PL_SZ                              0x51       81
[    0xc] CFG_PCIE_MAX_RDRQ_SZ                            0x52       82
[   0x10] CFG_SYS_ID                                      0x1234     4660
[   0x14] CFG_MSI_EN                                      0x2020202  33686018
[   0x18] CFG_PCIE_DATA_WIDTH                             0x3        3
[   0x1c] CFG_PCIE_CTRL                                   0x1        1
[   0x40] CFG_AXI_USR_MAX_PL_SZ                           0x55       85
[   0x44] CFG_AXI_USR_MAX_RDRQ_SZ                         0x55       85
[   0x4c] CFG_MISC_CTRL                                   0x10009    65545
[   0x80] CFG_SCRATCH_REG_0                               0          0
[   0x84] CFG_SCRATCH_REG_1                               0          0
[   0x88] CFG_SCRATCH_REG_2                               0          0
[   0x8c] CFG_SCRATCH_REG_3                                0          0
[   0x90] CFG_SCRATCH_REG_4                                0          0
[   0x94] CFG_SCRATCH_REG_5                                0          0
[   0x98] CFG_SCRATCH_REG_6                                0          0
[   0x9c] CFG_SCRATCH_REG_7                                0          0
[   0xf0] QDMA_RAM_SBE_MSK_A                               0xffffff11 4294967057
[   0xf4] QDMA_RAM_SBE_STS_A                               0          0
[   0xf8] QDMA_RAM_DBE_MSK_A                               0xffffff11 4294967057
[   0xfc] QDMA_RAM_DBE_STS_A                               0          0
[  0x100] GLBL2_ID                                         0x1fd70000 534183936
[  0x104] GLBL2_PF_BL_INT                                  0x41041    266305
[  0x108] GLBL2_PF_VF_BL_INT                               0          0
[  0x10c] GLBL2_PF_BL_EXT                                  0x104104   1065220
[  0x110] GLBL2_PF_VF_BL_EXT                               0          0
[  0x114] GLBL2_CHNL_INST                                  0x30101    196865
[  0x118] GLBL2_CHNL_QDMA                                  0x30f0f    200463
[  0x11c] GLBL2_CHNL_STRM                                  0x30000    196608
[  0x120] GLBL2_QDMA_CAP                                   0x800      2048
[  0x128] GLBL2_PASID_CAP                                  0          0
[  0x12c] GLBL2_FUNC_RET                                   0          0
[  0x130] GLBL2_SYS_ID                                     0          0
[  0x134] GLBL2_MISC_CAP                                   0x1000000  16777216
[  0x1b8] GLBL2_DBG_PCIE_RQ_0                              0x7f50003  133496835
[  0x1bc] GLBL2_DBG_PCIE_RQ_1                              0x6024     24612
[  0x1c0] GLBL2_DBG_AXIMM_WR_0                             0x600021   6291489
[  0x1c4] GLBL2_DBG_AXIMM_WR_1                             0          0
[  0x1c8] GLBL2_DBG_AXIMM_RD_0                             0x1        1
[  0x1cc] GLBL2_DBG_AXIMM_RD_1                             0          0
[  0x204] GLBL_RNGSZ_0                                     0x801      2049
[  0x208] GLBL_RNGSZ_1                                     0x41       65
[  0x20c] GLBL_RNGSZ_2                                     0x81       129
[  0x210] GLBL_RNGSZ_3                                     0xc1       193
[  0x214] GLBL_RNGSZ_4                                     0x101      257
[  0x218] GLBL_RNGSZ_5                                     0x181      385
[  0x21c] GLBL_RNGSZ_6                                     0x201      513
[  0x220] GLBL_RNGSZ_7                                     0x301      769
[  0x224] GLBL_RNGSZ_8                                     0x401      1025
[  0x228] GLBL_RNGSZ_9                                     0x601      1537
[  0x22c] GLBL_RNGSZ_10                                    0xc01      3073
[  0x230] GLBL_RNGSZ_11                                    0x1001     4097
[  0x234] GLBL_RNGSZ_12                                    0x1801     6145
[  0x238] GLBL_RNGSZ_13                                    0x2001     8193
[  0x23c] GLBL_RNGSZ_14                                    0x3001     12289
[  0x240] GLBL_RNGSZ_15                                    0x4001     16385
[  0x248] GLBL_ERR_STAT                                    0          0
[  0x24c] GLBL_ERR_MASK                                    0x90f      2319
[  0x250] GLBL_DSC_CFG                                     0x35       53
[  0x254] GLBL_DSC_ERR_STS                                 0          0
[  0x258] GLBL_DSC_ERR_MSK                                 0x1f9023f  33096255
[  0x25c] GLBL_DSC_ERR_LOG_0                               0          0
[  0x260] GLBL_DSC_ERR_LOG_1                               0          0
[  0x264] GLBL_TRQ_ERR_STS                                 0          0
[  0x268] GLBL_TRQ_ERR_MSK                                 0xf        15
[  0x26c] GLBL_TRQ_ERR_LOG                                 0          0
[  0x270] GLBL_DSC_DBG_DAT_0                               0          0
[  0x274] GLBL_DSC_DBG_DAT_1                               0x8080     32896
[  0x27c] GLBL_DSC_ERR_LOG2                                0          0
[  0x288] GLBL_INTERRUPT_CFG                               0          0
[  0x400] TRQ_SEL_FMAP_0                                   0x100000   1048576
[  0x404] TRQ_SEL_FMAP_1                                   0          0
[  0x408] TRQ_SEL_FMAP_2                                   0          0
[  0x40c] TRQ_SEL_FMAP_3                                   0          0
[  0x804] IND_CTXT_DATA_0                                  0          0
[  0x808] IND_CTXT_DATA_1                                  0          0
[  0x80c] IND_CTXT_DATA_2                                  0          0
[  0x810] IND_CTXT_DATA_3                                  0          0
[  0x814] IND_CTXT_DATA_4                                  0          0
[  0x818] IND_CTXT_DATA_5                                  0          0
[  0x81c] IND_CTXT_DATA_6                                  0          0
[  0x820] IND_CTXT_DATA_7                                  0          0
[  0x824] IND_CTXT_MASK_0                                  0xffffffff 4294967295
[  0x828] IND_CTXT_MASK_1                                  0xffffffff 4294967295
[  0x82c] IND_CTXT_MASK_2                                  0xffffffff 4294967295
[  0x830] IND_CTXT_MASK_3                                  0xffffffff 4294967295
[  0x834] IND_CTXT_MASK_4                                  0xffffffff 4294967295
[  0x838] IND_CTXT_MASK_5                                  0xffffffff 4294967295
[  0x83c] IND_CTXT_MASK_6                                  0xffffffff 4294967295
[  0x840] IND_CTXT_MASK_7                                  0xffffffff 4294967295
[  0x844] IND_CTXT_CMD                                     0xa2       162
[  0xa00] C2H_TIMER_CNT_0                                  0x1        1
[  0xa04] C2H_TIMER_CNT_1                                  0x2        2
[  0xa08] C2H_TIMER_CNT_2                                  0x4        4
[  0xa0c] C2H_TIMER_CNT_3                                  0x5        5
[  0xa10] C2H_TIMER_CNT_4                                  0x8        8
[  0xa14] C2H_TIMER_CNT_5                                  0xa        10
[  0xa18] C2H_TIMER_CNT_6                                  0xf        15
[  0xa1c] C2H_TIMER_CNT_7                                  0x14       20
[  0xa20] C2H_TIMER_CNT_8                                  0x19       25
[  0xa24] C2H_TIMER_CNT_9                                  0x1e       30
[  0xa28] C2H_TIMER_CNT_10                                 0x32       50
[  0xa2c] C2H_TIMER_CNT_11                                 0x4b       75
[  0xa30] C2H_TIMER_CNT_12                                 0x64       100
[  0xa34] C2H_TIMER_CNT_13                                 0x7d       125
[  0xa38] C2H_TIMER_CNT_14                                 0x96       150
[  0xa3c] C2H_TIMER_CNT_15                                 0xc8       200
[  0xa40] C2H_CNT_THRESH_0                                 0x40       64
[  0xa44] C2H_CNT_THRESH_1                                 0x2        2
[  0xa48] C2H_CNT_THRESH_2                                 0x4        4
[  0xa4c] C2H_CNT_THRESH_3                                0x8        8
[  0xa50] C2H_CNT_THRESH_4                                0x10       16
[  0xa54] C2H_CNT_THRESH_5                                0x18       24
[  0xa58] C2H_CNT_THRESH_6                                0x20       32
[  0xa5c] C2H_CNT_THRESH_7                                0x30       48
[  0xa60] C2H_CNT_THRESH_8                                0x50       80
[  0xa64] C2H_CNT_THRESH_9                                0x60       96
[  0xa68] C2H_CNT_THRESH_10                               0x70       112
[  0xa6c] C2H_CNT_THRESH_11                               0x80       128
[  0xa70] C2H_CNT_THRESH_12                               0x90       144
[  0xa74] C2H_CNT_THRESH_13                               0xa0       160
[  0xa78] C2H_CNT_THRESH_14                               0xb0       176
[  0xa7c] C2H_CNT_THRESH_15                               0xc0       192
[  0xa88] C2H_STAT_S_AXIS_C2H_ACCEPTED                    0          0
[  0xa8c] C2H_STAT_S_AXIS_CMPT_ACCEPTED                   0          0
[  0xa90] C2H_STAT_DESC_RSP_PKT_ACCEPTED                  0          0
[  0xa94] C2H_STAT_AXIS_PKG_CMP                           0          0
[  0xa98] C2H_STAT_DESC_RSP_ACCEPTED                      0          0
[  0xa9c] C2H_STAT_DESC_RSP_CMP                           0          0
[  0xaa0] C2H_STAT_WRQ_OUT                                0          0
[  0xaa4] C2H_STAT_WPL_REN_ACCEPTED                       0          0
[  0xaa8] C2H_STAT_TOTAL_WRQ_LEN                          0          0
[  0xaac] C2H_STAT_TOTAL_WPL_LEN                          0          0
[  0xab0] C2H_BUF_SZ_0                                    0x1000     4096
[  0xab4] C2H_BUF_SZ_1                                    0x100      256
[  0xab8] C2H_BUF_SZ_2                                    0x200      512
[  0xabc] C2H_BUF_SZ_3                                    0x400      1024
[  0xac0] C2H_BUF_SZ_4                                    0x800      2048
[  0xac4] C2H_BUF_SZ_5                                    0xf80      3968
[  0xac8] C2H_BUF_SZ_6                                    0x1000     4096
[  0xacc] C2H_BUF_SZ_7                                    0x1000     4096
[  0xad0] C2H_BUF_SZ_8                                    0x1000     4096
[  0xad4] C2H_BUF_SZ_9                                    0x1000     4096
[  0xad8] C2H_BUF_SZ_10                                   0x1000     4096
[  0xadc] C2H_BUF_SZ_11                                   0x1000     4096
[  0xae0] C2H_BUF_SZ_12                                   0x1000     4096
[  0xae4] C2H_BUF_SZ_13                                   0x2000     8192
[  0xae8] C2H_BUF_SZ_14                                   0x233a     9018
[  0xaec] C2H_BUF_SZ_15                                   0x4000     16384
[  0xaf0] C2H_ERR_STAT                                    0          0
[  0xaf4] C2H_ERR_MASK                                    0xfedb     65243
[  0xaf8] C2H_FATAL_ERR_STAT                              0          0
[  0xafc] C2H_FATAL_ERR_MASK                              0x7df1b    515867
[  0xb00] C2H_FATAL_ERR_ENABLE                            0          0
[  0xb04] GLBL_ERR_INT                                    0x1002000  16785408
[  0xb08] C2H_PFCH_CFG                                    0xc201100  203428096
[  0xb0c] C2H_INT_TIMER_TICK                              0x19       25
[  0xb10] C2H_STAT_DESC_RSP_DROP_ACCEPTED                 0          0
[  0xb14] C2H_STAT_DESC_RSP_ERR_ACCEPTED                  0          0
[  0xb18] C2H_STAT_DESC_REQ                               0          0
[  0xb1c] C2H_STAT_DEBUG_DMA_ENG_0                        0          0
[  0xb20] C2H_STAT_DEBUG_DMA_ENG_1                        0x80000000 2147483648
[  0xb24] C2H_STAT_DEBUG_DMA_ENG_2                        0xc0000000 3221225472
[  0xb28] C2H_STAT_DEBUG_DMA_ENG_3                        0          0
[  0xb2c] C2H_DBG_PFCH_ERR_CTXT                           0x1d003    118787
[  0xb30] C2H_FIRST_ERR_QID                               0          0
[  0xb34] STAT_NUM_CMPT_IN                                0          0
[  0xb38] STAT_NUM_CMPT_OUT                               0          0
[  0xb3c] STAT_NUM_CMPT_DRP                               0          0
[  0xb40] STAT_NUM_STAT_DESC_OUT                          0          0
[  0xb44] STAT_NUM_DSC_CRDT_SENT                          0          0
[  0xb48] STAT_NUM_FCH_DSC_RCVD                           0          0
[  0xb4c] STAT_NUM_BYP_DSC_RCVD                           0          0
[  0xb50] C2H_CMPT_COAL_CFG                               0x40064014 1074151444
[  0xb54] C2H_INTR_H2C_REQ                                0x3        3
[  0xb58] C2H_INTR_C2H_MM_REQ                             0x1        1
[  0xb5c] C2H_INTR_ERR_INT_REQ                            0          0
[  0xb60] C2H_INTR_C2H_ST_REQ                             0          0
[  0xb64] C2H_INTR_H2C_ERR_MM_MSIX_ACK                    0x4        4
[  0xb68] C2H_INTR_H2C_ERR_MM_MSIX_FAIL                   0          0
[  0xb6c] C2H_INTR_H2C_ERR_MM_NO_MSIX                     0          0
[  0xb70] C2H_INTR_H2C_ERR_MM_CTXT_INVAL                  0          0
[  0xb74] C2H_INTR_C2H_ST_MSIX_ACK                        0          0
[  0xb78] C2H_INTR_C2H_ST_MSIX_FAIL                       0          0
[  0xb7c] C2H_INTR_C2H_ST_NO_MSIX                         0          0
[  0xb80] C2H_INTR_C2H_ST_CTXT_INVAL                      0          0
[  0xb84] C2H_STAT_WR_CMP                                 0          0
[  0xb88] C2H_STAT_DEBUG_DMA_ENG_4                        0x40000000 1073741824
[  0xb8c] C2H_STAT_DEBUG_DMA_ENG_5                        0          0
[  0xb90] C2H_DBG_PFCH_QID                                0          0
[  0xb94] C2H_DBG_PFCH                                    0          0
[  0xb98] C2H_INT_DEBUG                                   0          0
[  0xb9c] C2H_STAT_IMM_ACCEPTED                           0          0
[  0xba0] C2H_STAT_MARKER_ACCEPTED                        0          0
[  0xba4] C2H_STAT_DISABLE_CMP_ACCEPTED                   0          0
[  0xba8] C2H_C2H_PAYLOAD_FIFO_CRDT_CNT                   0          0
[  0xbac] C2H_INTR_DYN_REQ                                0x4        4
[  0xbb0] C2H_INTR_DYN_MSIX                               0          0
[  0xbb4] C2H_DROP_LEN_MISMATCH                           0          0
[  0xbb8] C2H_DROP_DESC_RSP_LEN                           0          0
[  0xbbc] C2H_DROP_QID_FIFO_LEN                           0          0
[  0xbc0] C2H_DROP_PAYLOAD_CNT                            0          0
[  0xbc4] QDMA_C2H_CMPT_FORMAT_0                          0x20001    131073
[  0xbc8] QDMA_C2H_CMPT_FORMAT_1                          0          0
[  0xbcc] QDMA_C2H_CMPT_FORMAT_2                          0          0
[  0xbd0] QDMA_C2H_CMPT_FORMAT_3                          0          0
[  0xbd4] QDMA_C2H_CMPT_FORMAT_4                          0          0
[  0xbd8] QDMA_C2H_CMPT_FORMAT_5                          0          0
[  0xbdc] QDMA_C2H_CMPT_FORMAT_6                          0          0
[  0xbe0] C2H_PFCH_CACHE_DEPTH                            0x10       16
[  0xbe4] C2H_CMPT_COAL_BUF_DEPTH                         0x10       16
[  0xbe8] C2H_PFCH_CRDT                                   0          0
[  0xe00] H2C_ERR_STAT                                    0          0
[  0xe04] H2C_ERR_MASK                                    0x1f       31
[  0xe08] H2C_FIRST_ERR_QID                               0          0
[  0xe0c] H2C_DBG_REG_0                                   0x10001    65537
[  0xe10] H2C_DBG_REG_1                                   0x20010    131088
[  0xe14] H2C_DBG_REG_2                                   0          0
[  0xe18] H2C_DBG_REG_3                                   0x44008025 1140883493
[  0xe1c] H2C_DBG_REG_4                                   0x22119000 571576320
[  0xe20] H2C_FATAL_ERR_EN                                0          0
[  0xe24] H2C_REQ_THROT                                   0xc14000   12664832
[  0xe28] H2C_ALN_DBG_REG0                                0x1        1
[ 0x1004] C2H_MM_CONTROL_0                                0x1        1
[ 0x1008] C2H_MM_CONTROL_1                                0x1        1
[ 0x100c] C2H_MM_CONTROL_2                                0x1        1
[ 0x1040] C2H_MM_STATUS_0                                 0x1        1
[ 0x1044] C2H_MM_STATUS_1                                 0x1        1
[ 0x1048] C2H_MM_CMPL_DSC_CNT                             0x1        1
[ 0x1054] C2H_MM_ERR_CODE_EN_MASK                         0          0
[ 0x1058] C2H_MM_ERR_CODE                                 0          0
[ 0x105c] C2H_MM_ERR_INFO                                 0          0
[ 0x10c0] C2H_MM_PERF_MON_CTRL                            0          0
[ 0x10c4] C2H_MM_PERF_MON_CY_CNT_0                        0          0
[ 0x10c8] C2H_MM_PERF_MON_CY_CNT_1                        0          0
[ 0x10cc] C2H_MM_PERF_MON_DATA_CNT_0                      0          0
[ 0x10d0] C2H_MM_PERF_MON_DATA_CNT_1                      0          0
[ 0x10e8] C2H_MM_DBG_INFO_0                               0x10002    65538
[ 0x10ec] C2H_MM_DBG_INFO_1                               0          0
[ 0x1204] H2C_MM_CONTROL_0                                0x1        1
[ 0x1208] H2C_MM_CONTROL_1                                0x1        1
[ 0x120c] H2C_MM_CONTROL_2                                0x1        1
[ 0x1240] H2C_MM_STATUS                                   0x1        1
[ 0x1248] H2C_MM_CMPL_DSC_CNT                             0x2        2
[ 0x1254] H2C_MM_ERR_CODE_EN_MASK                         0          0
[ 0x1258] H2C_MM_ERR_CODE                                 0          0
[ 0x125c] H2C_MM_ERR_INFO                                 0          0
[ 0x12c0] H2C_MM_PERF_MON_CTRL                            0          0
[ 0x12c4] H2C_MM_PERF_MON_CY_CNT_0                        0          0
[ 0x12c8] H2C_MM_PERF_MON_CY_CNT_1                        0          0
[ 0x12cc] H2C_MM_PERF_MON_DATA_CNT_0                      0          0
[ 0x12d0] H2C_MM_PERF_MON_DATA_CNT_1                      0          0
[ 0x12e8] H2C_MM_DBG_INFO                                 0x10002    65538
[ 0x12ec] H2C_MM_REQ_THROT                                0x8000     32768
```

Output of a queue parameters dump

```
qdma17000-MM-0 C2H online
          hw_ID 0, thp ?, desc 0xffff880035850000/0x35850000, 1536

          cmpl status: 0xffff88003585c000, 00010000 00000001
          SOFTWARE CTXT:
                    Interrupt Aggregation                           0x1        1
                    Ring Index                                      0x0        0
                    Descriptor Ring Base Addr (High)                0x0        0
                    Descriptor Ring Base Addr (Low)                 0x35850000 897908736
                    Is Memory Mapped                                0x1        1
                    Marker Disable                                  0x0        0
                    IRQ Request                                     0x0        0
                    Writeback Error Sent                            0x0        0
                    Error                                           0x0        0
                    Interrupt No Last                               0x0        0
                    Port Id                                         0x0        0
                    Interrupt Enable                                0x1        1
                    Writeback Enable                                0x1        1
                    MM Channel                                      0x0        0
                    Bypass Enable                                   0x0        0
                    Descriptor Size                                 0x2        2
                    Ring Size                                       0x9        9
                    Fetch Max                                       0x0        0
                    Address Translation                             0x0        0
                    Write back/Intr Interval                        0x1        1
                    Write back/Intr Check                           0x1        1
                    Fetch Credit Enable                             0x0        0
                    Queue Enable                                    0x1        1
                    Function Id                                     0x0        0
                    IRQ Arm                                         0x1        1
                    PIDX                                            0x1        1

          HARDWARE CTXT:
                    Fetch Pending                                   0x0        0
                    Eviction Pending                                0x0        0
                    Queue Invalid No Desc Pending                   0x1        1
                    Descriptors Pending                             0x0        0
                    Credits Consumed                                0x0        0
                    CIDX                                            0x1        1

          CREDIT CTXT:
                    Credit                                          0x0        0

          INTR CTXT:
                    at                                              0x0        0
                    pidx                                            0x4        4
                    page_size                                       0x0        0
                    baddr_4k (High)                                 0x0        0
                    baddr_4k (Low)                                  0x35b39000 900960256
                    color                                           0x1        1
                    int_st                                          0x0        0
                    vec                                             0x3        3
                    valid                                           0x1        1

          total descriptor processed:    1
qdma17000-MM-0 H2C online
          hw_ID 0, thp ?, desc 0xffff880042880000/0x42880000, 1536

          cmpl status: 0xffff88004288c000, 00020000 00000002
          SOFTWARE CTXT:
                    Interrupt Aggregation                           0x1        1
                    Ring Index                                      0x0        0
                    Descriptor Ring Base Addr (High)                0x0        0
                    Descriptor Ring Base Addr (Low)                 0x42880000 1116209152
                    Is Memory Mapped                                0x1        1
                    Marker Disable                                  0x0        0
                    IRQ Request                                     0x0        0
                    Writeback Error Sent                            0x0        0
                    Error                                           0x0        0
                    Interrupt No Last                               0x0        0
                    Port Id                                         0x0        0
                    Interrupt Enable                                0x1        1
                    Writeback Enable                                0x1        1
                    MM Channel                                      0x0        0
                    Bypass Enable                                   0x0        0
                    Descriptor Size                                 0x2        2
                    Ring Size                                       0x9        9
                    Fetch Max                                       0x0        0
                    Address Translation                             0x0        0
                    Write back/Intr Interval                        0x1        1
                    Write back/Intr Check                           0x1        1
                    Fetch Credit Enable                             0x0        0
                    Queue Enable                                    0x1        1
                    Function Id                                     0x0        0
                    IRQ Arm                                         0x1        1
                    PIDX                                            0x2        2

          HARDWARE CTXT:
                    Fetch Pending                                   0x0        0
                    Eviction Pending                                0x0        0
                    Queue Invalid No Desc Pending                   0x1        1
                    Descriptors Pending                             0x0        0
                    Credits Consumed                                0x0        0
                    CIDX                                            0x2        2

          CREDIT CTXT:
                    Credit                                          0x0        0

          INTR CTXT:
                    at                                              0x0        0
                    pidx                                            0x4        4
                    page_size                                       0x0        0
                    baddr_4k (High)                                 0x0        0
                    baddr_4k (Low)                                  0x35b39000 900960256
                    color                                           0x1        1
                    int_st                                          0x0        0
                    vec                                             0x3        3
                    valid                                           0x1        1

          total descriptor processed:    2
Dumped Queues 0 -> 0.
```

Revision History

09/30/2019 - Initial release