PCIe DMA Demo

User Guide

2023-06-09

Disclaimer

GOWINSEMI assumes no liability and provides no warranty (either expressed or implied) and is not responsible for any damage incurred to your hardware, software, data, or property resulting from usage of the materials or intellectual property except as outlined in the GOWINSEMI Terms and Conditions of Sale. GOWINSEMI may make changes to this document at any time without prior notice. Anyone relying on this documentation should contact GOWINSEMI for the current documentation and errata.

Revision History

Date	Version	Description
06/09/2023	1.0	Initial version published.

Contents

1 About This Guide

1.1 Purpose

This document has been written for designers of FPGAs who are evaluating or using the Arora V PCI Express Controller IP. The Gowin PCIe DMA demo demonstrates a simple DMA implementation based on the Arora V PCI Express Controller IP. The DEMO contains the complete FPGA source code and project files for the DK_START_GW5AT-LV138FPG676A_V2.0 development board. Drivers and applications for the Linux operating system are also included.

1.2 Related Documents

The latest user guides are available on the GOWINSEMI website. You can find the related documents at www.gowinsemi.com:

- DS1104，GW5AST series of FPGA Products Data Sheet
- IPUG1020，Arora V PCI Express Controller IP User Guide
- SUG100，Gowin EDA Software User Guide

1.3 Terminology and Abbreviation

The terminology and abbreviations used in this manual are as shown in Table 1-1.

Table 1-1 Terminology and Abbreviations

Terminology and Abbreviations	Full Name
PCIe	Peripheral Component Interconnect Express
DMA	Direct Memory Access
BAR	Base Address Register
C2H	Card to Host
H2C	Host to Card

1.4 Support and Feedback

Gowin Semiconductor provides customers with comprehensive technical support. If you have any questions, comments, or suggestions, please feel free to contact us directly using the information provided below.

Website：www.gowinsemi.com.cn

E-mail：support@gowinsemi.com

Tel：+86-755-8262-0391

2 Overview

PCIe DMA Demo is a PCIe demonstration DMA application on the DK_START_GW5AT-LV138FPG676A_V2.0 development board. The demo is based on the Arora V PCI Express Controller IP, plus BAR read and write logic, DMA control logic, cache management logic and interrupt control logic to form the main body of the FPGA project, which is written in verilog language. This demo also contains the source code of simple drivers and applications under the Linux system for reference.

The main feature of PCIe DMA Demo are as follows：

- Compliant to the PCI Express Base Specification 2.0
- Supports x4 lanes
- Supports one channel H2C DMA and one channel C2H DMA
- Supports simultaneous H2C and C2H DMA transmission
- Uses one BAR for DMA control
- With Linux driver example
- With Linux application example

PCIe DMA Demo includes three parts:

- pcie_dma_demo.gar - FPGA project archive file
- pcie_dma_demo_fs.zip - compressed file for bitstream
- pcie_dma_demo.tar.gz - compressed file for driver and application

3 Instructions

3.1 FPGA project construction

Use the restore archived project function of Gowin EDA software to restore the archived project file pcie_dma_demo.gar。

Clicking “Restore Archived Project” in “Project” in the menu bar will bring up the Restore Archived Project dialog box, as shown in Figure 3-1.

Figure 3-1 Restore Archived Project dialog box

Click the “Archived File” path selection button to select the pcie_dma_demo.gar file. You can use the default path or modify it as needed, then click the “OK” button to complete the project recovery.。

3.2 Hardware environment construction

The hardware environment needs to prepare a PC host with a free PCI Express slot, in order to maximize the performance of the demo, the further requirements for the PCI Express slot are: not less than x4 lanes, and support PCI Express 2.0. Typically, a PC host with integrated graphics will have a free x16 PCI Express slot.

As shown in Figure 3-2, the x16 PCI Express slot is marked in the red box, the x1 PCI Express slot is marked in the yellow box, and the PCI slot is marked in the blue box.

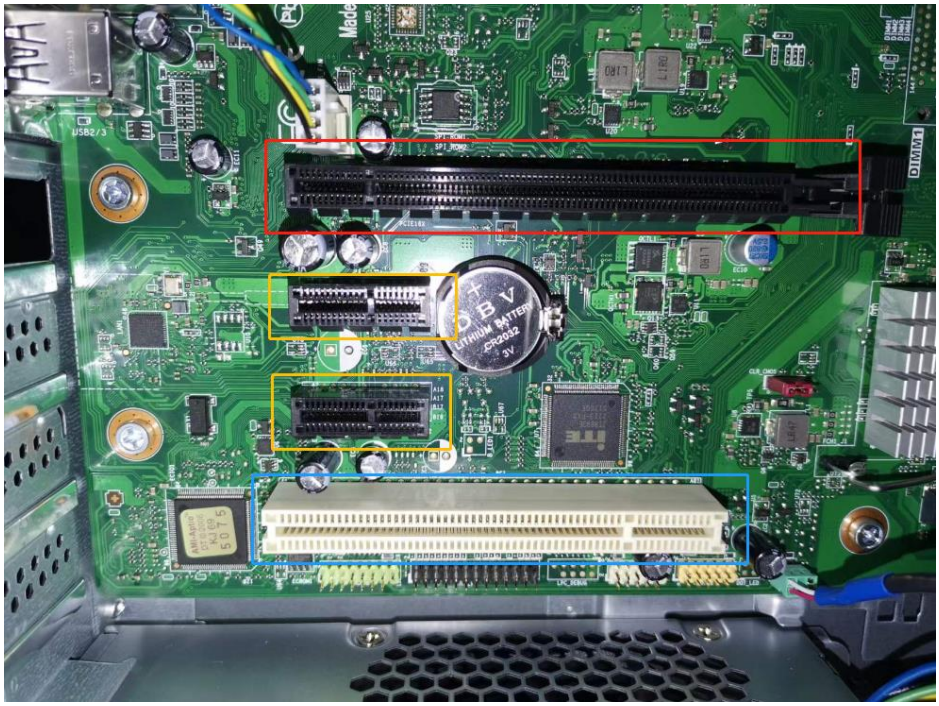

Figure 3-2 PCI Express slot illustration

Due to the relatively large size of the V 2.0 development board, it may not be possible to insert directly into the PCI Express slot, and an additional PCI Express extension cable need to be prepared to build the hardware environment. Figure 3-3 shows a PCI Express x16 extension cable actually used in the lab. You can choose to purchase by searching for the keyword "PCIE+x16+Cable" in the amazon.com.

Figure 3-3 PCI Express x16 extension cable

Figure 3-4 shows a picture of the development board connected to a PC with the PCI Express extension cable.

Figure 3-4 PC, cable and development board

3.3 Linux Environment construction

3.3.1 Install Linux

The first is the Linux system installation, here it is recommended to install Ubuntu 20.04 and above.

The Chinese reference for Ubuntu 20.04 installation：
https://www.sysgeek.cn/install-ubuntu-20-04-lts-desktop/

The English reference for Ubuntu 22.04 installation：
https://www.makeuseof.com/how-to-install-ubuntu-on-your-pc/

3.3.2 Modify Boot Arguments

After the Linux system is installed, you need to manually modify the relevant boot arguments of iommu, otherwise reading and writing system memory in the demo will easily cause system brokendown. The following are the modification steps and confirmation methods:

- Use vi to modify /etc/default/grub. Add "iommu=pt" to GRUB_CMDLINE_LINUX_DEFAULT:

```
$ sudo vi /etc/default/grub
```

Modify

```
…
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""
```

to：

```
…
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash iommu=pt"
GRUB_CMDLINE_LINUX=""
```

Then save and exit。

- Run the following command to update the GRUB boot arguments:

```
$ sudo update-grub
```

- After rebooting the system, check the results with the following command:

```
$ cat /proc/cmdline
```

You should see the words “iommu=pt” added earlier. For example：

```
BOOT_IMAGE=/boot/vmlinuz-5.15.0-73-generic root=UUID=666e15fc-916e-4ec4-86be-0d991eb10b9a ro quiet splash iommu=pt vt.handoff=7
```

3.3.3 Preparation tools and source code

Nex, install the toolchain required to compile the driver and application, use the shortcut key CTRL+ALT+t to open the terminal, enter the installation command, and complete the installation:

```
$ sudo apt install make gcc
```

If you have upgraded the kernel for Ubuntu, you will need to re-download and install the kernel header files required to compile the driver. Installation command：

```
$ sudo apt install linux-headers-$(uname -r)
```

Next, after copying the driver and application package, complete the extraction from the command line：

```
$ tar xvf gowin_pcie_demo.tar.gz
```

Enter the source code directory and execute the make command to

complete the compilation of the driver and application：

```
$ cd gowin_pcie_demo
$ make
```

3.4 Execute the test script

Run the demo script with the following command, select the test mode and block size as prompted, and complete the test.

```
$ sudo ./bin/gowin_demo
```

```
…
Found PCIe device:
01:00.0 Memory controller: Gowin Semiconductor Corporation Device 1100

* Test mode:
1. Performance test for copy_to_host
2. Performance test for copy_to_card
Please enter your choice: 1 or 2: 1
* Block size (bytes):
1. 256  2. 512  3. 1024  4. 2048  5. 4096  6. 8192
Please enter your choice: 1 ~ 6: 5

Total data: 512000000 bytes.
Time elapsed: 301267503 ns.
Speed: 13.596 Gbps (85.0%)
```

Actions such as driver detection and loading are already included in ‘gowin_demo’ script and do not need to be handled separately. If needed, you can check the device status through the ‘lspci’ command and load/unload the driver through the ‘insmod/rmmod’ command.

- Check device

```
$ sudo lspci -vvd 22c2:1100
```

If the board is properly connected and the bitstream is loaded correctly, you will see the details of the PCIe device, otherwise there will be no information. For example：

```
01:00.0 Memory controller: Gowin Semiconductor Corporation Device 1100
        Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx+
        Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx+
        Latency: 0
        Interrupt: pin A routed to IRQ 108
        IOMMU group: 8
        Region 0: Memory at c1000000 (64-bit, prefetchable) [size=1M]
        Region 2: Memory at c1200000 (64-bit, prefetchable) [size=64K]
        Region 4: Memory at c1100000 (64-bit, prefetchable) [size=1M]
        Capabilities: [80] Express (v2) Legacy Endpoint, MSI 1f
                DevCap: MaxPayload 1024 bytes, PhantFunc 0, Latency L0s <64ns, L1 <1us
                        ExtTag+ AttnBtn- AttnInd- PwrInd- RBE+ FLReset-
                DevCtl: CorrErr- NonFatalErr- FatalErr- UnsupReq-
                        RlxdOrd+ ExtTag+ PhantFunc- AuxPwr- NoSnoop+
                        MaxPayload 512 bytes, MaxReadReq 4096 bytes
    …
                LnkSta: Speed 5GT/s (ok), Width x4 (downgraded)
    …
        Kernel driver in use: gowin_pcie_demo
```

- Load the driver

```
$ sudo insmod ./driver/gowin_demo.ko
```

The ‘lsmod’ command can be used to check if the driver is loaded:

```
$ lsmod | grep gowin
gowin_demo                32768  0
```

- Unload the driver

```
$ sudo rmmod gowin_demo
```


GOWIN
PROGRAMMING FOR THE FUTURE