

Xilinx Wiki

SHORTCUTS

Xilinx Wiki Home

Xilinx.com

Pages

▸ Linux Prebuilt Images

▸ Versal ACAPs

▸ Zynq-7000

▸ Zynq UltraScale+ MPSoC

▸ Zynq UltraScale+ RFSoc

▸ MicroBlaze

▾ Linux

▸ PetaLinux

▸ Yocto


▸ Xilinx Open Source Linux

▸ Real-Time Linux

▾ Linux Drivers

📖 Archived pages

Xilinx Wiki / ... / Zynq UltraScale+ MPSoC PS-PCle End Point Driver

 Last updated: Apr 20, 2022 · 4 min read

This page gives an overview of how to use the Linux device driver for the Xilinx Zynq UltraScale+ MPSoC PS PCIe End Point DMA functionality.

The driver runs on the host machine on which the end point is connected. The driver DMA and PIO functionality on the End Point can be tested using an application. Details are provided below on the required setup, how to compile the driver, applications and how to test the DMA and PIO functionality on the End Point.

Table of Contents

- [Controller for PCI Express DMA](#)
- [Hardware Setup](#)
- [Software](#)
- [Set up the Host machine \(x86 Ubuntu\)](#)
- [Testing](#)
- [Related Links](#)

Controller for PCI Express DMA

PCI Express (abbreviated as PCIe) is the newest bus standard designed to replace the old PCI/PCI-X and AGP standards. PCIe is used in servers, consumer, and industrial applications either as a motherboard-level interconnection to link peripherals or as an expansion card interface for add-on boards. Zynq® UltraScale+™ MPSoC devices provide a controller for the integrated block for PCI Express® v2.1 compliant, AXI-PCIe bridge, and DMA modules. The AXI-PCIe bridge provides high-performance bridging between PCIe and AXI.

The controller for PCIe supports both Endpoint and Root Port modes of operations and provides support for up to x4 Gen2 links.

The DMA controller for PCIe contains a high-performance 4-channel direct memory access (DMA) engine. Each channel can be programmed for either transmit or receive DMA operation. Each channel can be controlled from either the PCIe or AXI domains. The DMA supports separate source and destination scatter-gather queues. The DMA hardware is responsible for merging the source and destination information for data movement. The scatter-gather elements can be located in either PCIe or AXI memory.

For more information about the controller for PCI Express, please refer to the Zynq UltraScale+ MPSoC TRM (UG1085).

Hardware Setup

- x86_64 host system with at least one Gen 2x4 PCIe slot.
- The End Point is tested with an x86 machine with Ubuntu 20.04 LTS installed.
- Xilinx Zynq UltraScale+ MPSoC PS PCIe EndPoint Board connected on an x86 machine's PCIe Slot.

Software

The DMA driver is available at [GitHub - Xilinx/zynqmp-pspcie-epdma](#) for reference.

📌 Ensure that the Device ID 0xA808 is configured in Vivado as the driver probe uses this device ID.

Set up the Host machine (x86 Ubuntu)

Power on and Open the terminal in the x86 Host system with Ubuntu and follow the below steps.

- OS

Verified drivers on Ubuntu 20.04.04 LTS

```
1 localhost@localhost:/home/local_host/zynqmp-pspcie-epdma$ lsb_release -a
2 No LSB modules are available.
3 Distributor ID: Ubuntu
4 Description:    Ubuntu 20.04.4 LTS
5 Release:       20.04
6 Codename:      focal
```

- Clone the Xilinx Zynq UltraScale+ MPSoC PS PCIe End point drivers by executing the below command:

```
1 $ git clone git@github.com:Xilinx/zynqmp-pspcie-epdma.git
```

- Change directory to the cloned directory:

```
1 $ cd zynqmp-pspcie-epdma
```

- Compile the Kernel Module Driver and application using the below command:

```
1 $ make
```

Testing

Program the Zynq MP as End Point board with the required boot files where the device is configured as the end point and then reboot the x86 Ubuntu Host system. Wait until boot.

Verify whether the device has been detected by the host system or not, by executing the below command in the host terminal.

```
1 $ lspci | grep -i "Xilinx"
2
3 06:00.0 Memory controller: Xilinx Corporation Device a808
```

Open the host Ubuntu Terminal and follow the below steps to test the PS PCIe using a simple-test DMA Application.

- Change directory to the cloned directory:

```
1 cd zynqmp_ep_ps_pcie_dma
```

- Insert the PS PCIe DMA drivers into the running kernel by executing the below commands:

```
1 $ sudo make insert
```

📌 The following tests are done without booting Linux on a Zynq MP EndPoint.

- Change the permission ps_pcie* nodes, so that the application can run without sudo permission:

```
1 $ sudo chmod 777 /dev/ps_pcie*
```

- Navigate to the apps directory in the terminal.

- Execute the DMA transfers application by providing command line parameters:

```
1 ./simple_test -c 0 -a 0x100000 -l 1024 -d s2c
2
3 ./simple_test -c 1 -a 0x100000 -l 1024 -d c2s
4
5 -c option specifies channel number
6 -a option specifies end point address
7 -l option specifies packet length
8 -d option specifies transfer direction. It can be either s2c or c2s
```

```
1 localhost@localhost:/home/local_host/zynqmp-pspcie-epdma/apps$ ./simple_test -c 0 -a 0x100000 -l 1024 -d s2c
2 write return value is 1024
3 Total time taken for transferring 1024 bytes of data is 252 micro seconds
4 localhost@localhost:/home/local_host/zynqmp-pspcie-epdma/apps$ ./simple_test -c 1 -a 0x100000 -l 1024 -d c2s
5 read return value is 1024
6 Received data is :: Data :: First Byte 5a Last Byte 5a
7 Total time taken for transferring 1024 bytes of data is 93 micro seconds
```

- Execute the PIO test application.

Pre-requisite:

Before executing the pci_pio_test app, the user needs to run the bare-metal application [embeddedsw/xpciepsu_ep_enable_example.c at master · Xilinx/embeddedsw](#) on the PCIe endpoint device.

This standalone application will set up end point bridge ingress translation on BAR2 that is translated to the PS-DDR on the endpoint.

📌 Current ingress translation size is 1MB, please refer to the baremetal application for more details.

- PIO Command line parameters:

```
1 ./pci_pio_test -o 0x0 -l 64
2
3 -o option specifies offset at PCI BAR 2.
4 -l option specifies length of data to be written and read back.
5 This application writes specified length of buffer with random data at the BAR offset and reads back from the same offset.
6 Data mismatch error along with offset at which mismatch is detected is printed if any discrepancy is present between buffer written to memory and buffer retrieved from memory.
```

```
1 localhost@localhost:/home/local_host/zynqmp-pspcie-epdma/apps$ ./pio_test -o 0x0 -l 64
2 PIO test done!
3 localhost@localhost:/home/local_host/zynqmp-pspcie-epdma/apps$
```

- Remove the driver by executing the below command:

```
1 $ sudo make remove
```

Related Links

- [UG1085](#)

Related pages

Linux ZynqMP PS-PCle Root Port Driver

Xilinx Wiki

Often read together

Linux Soft PCIe Driver

Xilinx Wiki

Organized together

Xilinx Linux PL PCIe Root Port

Xilinx Wiki

Often read together

👍 Be the first to add a reaction

© Copyright 2019 - 2022 Xilinx Inc. [Privacy Policy](#)