PCI Non-Transparent Bridge (NTB) Endpoint Function (EPF) User Guide

Author: Kishon Vijay Abraham I <kishon@ti.com>

This document is a guide to help users use pci-epf-ntb function driver and ntb_hw_epf host driver for NTB functionality. The list of steps to be followed in the host side and EP side is given below. For the hardware configuration and internals of NTB using configurable endpoints see Documentation/PCI/endpoint/pci-ntb-function.rst

Endpoint Device

Endpoint Controller Devices

For implementing NTB functionality at least two endpoint controller devices are required.

To find the list of endpoint controller devices in the system:

```
# ls /sys/class/pci_epc/
2900000.pcie-ep  2910000.pcie-ep
```

If PCI_ENDPOINT_CONFIGFS is enabled:

```
# ls /sys/kernel/config/pci_ep/controllers
2900000.pcie-ep  2910000.pcie-ep
```

Endpoint Function Drivers

To find the list of endpoint function drivers in the system:

```
# ls /sys/bus/pci-epf/drivers
pci_epf_ntb   pci_epf_ntb
```

If PCI_ENDPOINT_CONFIGFS is enabled:

```
# ls /sys/kernel/config/pci_ep/functions
pci_epf_ntb   pci_epf_ntb
```

Creating pci-epf-ntb Device

PCI endpoint function device can be created using the configfs. To create pci-epf-ntb device, the following commands can be used:

```
# mount -t configfs none /sys/kernel/config
# cd /sys/kernel/config/pci_ep/
# mkdir functions/pci_epf_ntb/func1
```

The "mkdir func1" above creates the pci-epf-ntb function device that will be probed by pci_epf_ntb driver.

The PCI endpoint framework populates the directory with the following configurable fields:

```
# ls functions/pci_epf_ntb/func1
baseclass_code    deviceid          msi_interrupts    pci-epf-ntb.0
progif_code       secondary         subsys_id         vendorid
cache_line_size   interrupt_pin     msix_interrupts   primary
revid             subclass_code     subsys_vendor_id
```

The PCI endpoint function driver populates these entries with default values when the device is bound to the driver. The pci-epf-ntb driver populates vendorid with 0xffff and interrupt_pin with 0x0001:

```
# cat functions/pci_epf_ntb/func1/vendorid
0xffff
# cat functions/pci_epf_ntb/func1/interrupt_pin
0x0001
```

Configuring pci-epf-ntb Device

The user can configure the pci-epf-ntb device using its configfs entry. In order to change the vendorid and the deviceid, the following commands can be used:

```
# echo 0x104c > functions/pci_epf_ntb/func1/vendorid
# echo 0xb00d > functions/pci_epf_ntb/func1/deviceid
```

In order to configure NTB specific attributes, a new sub-directory to func1 should be created:

```
# mkdir functions/pci_epf_ntb/func1/pci_epf_ntb.0/
```

The NTB function driver will populate this directory with various attributes that can be configured by the user:

```
# ls functions/pci_epf_ntb/func1/pci_epf_ntb.0/
db_count    mw1         mw2         mw3         mw4         num_mws
spad_count
```

A sample configuration for NTB function is given below:

```
# echo 4 > functions/pci_epf_ntb/func1/pci_epf_ntb.0/db_count
# echo 128 > functions/pci_epf_ntb/func1/pci_epf_ntb.0/spad_count
# echo 2 > functions/pci_epf_ntb/func1/pci_epf_ntb.0/num_mws
# echo 0x100000 > functions/pci_epf_ntb/func1/pci_epf_ntb.0/mw1
# echo 0x100000 > functions/pci_epf_ntb/func1/pci_epf_ntb.0/mw2
```

Binding pci-epf-ntb Device to EP Controller

NTB function device should be attached to two PCI endpoint controllers connected to the two hosts. Use the 'primary' and 'secondary' entries inside NTB function device to attach one PCI endpoint controller to primary interface and the other PCI endpoint controller to the secondary interface:

```
# ln -s controllers/2900000.pcie-ep/ functions/pci-epf-ntb/func1/primary
# ln -s controllers/2910000.pcie-ep/ functions/pci-epf-ntb/func1/secondary
```

Once the above step is completed, both the PCI endpoint controllers are ready to establish a link with the host.

Start the Link

In order for the endpoint device to establish a link with the host, the _start_ field should be populated with '1'. For NTB, both the PCI endpoint controllers should establish link with the host:

```
# echo 1 > controllers/2900000.pcie-ep/start
# echo 1 > controllers/2910000.pcie-ep/start
```

RootComplex Device

lspci Output

Note that the devices listed here correspond to the values populated in "Creating pci-epf-ntb Device" section above:

```
# lspci
0000:00:00.0 PCI bridge: Texas Instruments Device b00d
0000:01:00.0 RAM memory: Texas Instruments Device b00d
```

Using ntb_hw_epf Device

The host side software follows the standard NTB software architecture in Linux. All the existing client side NTB utilities like NTB Transport Client and NTB Netdev, NTB Ping Pong Test Client and NTB Tool Test Client can be used with NTB function device.

For more information on NTB see :doc:`Non-Transparent Bridge <../../driver-api/ntb>`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\PCI\endpoint\(linux-master)(Documentation)(PCI)(endpoint)pci-ntb-howto.rst, line 160); *backlink*

Unknown interpreted text role "doc".