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

This document is a guide to use the PCI Endpoint Framework in order to create endpoint controller driver, endpoint function driver, and using configfs interface to bind the function driver to the controller driver.

Introduction

Linux has a comprehensive PCI subsystem to support PCI controllers that operates in Root Complex mode. The subsystem has capability to scan PCI bus, assign memory resources and IRQ resources, load PCI driver (based on vendor ID, device ID), support other services like hot-plug, power management, advanced error reporting and virtual channels.

However the PCI controller IP integrated in some SoCs is capable of operating either in Root Complex mode or Endpoint mode. PCI Endpoint Framework will add endpoint mode support in Linux. This will help to run Linux in an EP system which can have a wide variety of use cases from testing or validation, co-processor accelerator, etc.

PCI Endpoint Core

The PCI Endpoint Core layer comprises 3 components: the Endpoint Controller library, the Endpoint Function library, and the configfs layer to bind the endpoint function with the endpoint controller.

PCI Endpoint Controller(EPC) Library

The EPC library provides APIs to be used by the controller that can operate in endpoint mode. It also provides APIs to be used by function driver/library in order to implement a particular endpoint function.

APIs for the PCI controller Driver

This section lists the APIs that the PCI Endpoint core provides to be used by the PCI controller driver.

- devm_pci_epc_create()/pci_epc_create()

  The PCI controller driver should implement the following ops:

  - write_header: ops to populate configuration space header
  - set_bar: ops to configure the BAR
  - clear_bar: ops to reset the BAR
  - alloc_addr_space: ops to allocate in PCI controller address space
  - free_addr_space: ops to free the allocated address space
  - raise_irq: ops to raise a legacy, MSI or MSI-X interrupt
  - start: ops to start the PCI link
  - stop: ops to stop the PCI link

  The PCI controller driver can then create a new EPC device by invoking devm_pci_epc_create()/pci_epc_create().

- devm_pci_epc_destroy()/pci_epc_destroy()

  The PCI controller driver can destroy the EPC device created by either devm_pci_epc_create() or pci_epc_create() using devm_pci_epc_destroy() or pci_epc_destroy().

- pci_epc_linkup()

  In order to notify all the function devices that the EPC device to which they are linked has established a link with the host, the PCI controller driver should invoke pci_epc_linkup().

- pci_epc_mem_init()

  Initialize the pci_epc_mem structure used for allocating EPC addr space.

- pci_epc_mem_exit()

  Cleanup the pci_epc_mem structure allocated during pci_epc_mem_init().

EPC APIs for the PCI Endpoint Function Driver

This section lists the APIs that the PCI Endpoint core provides to be used by the PCI endpoint function driver.

- pci_epc_write_header()

  The PCI endpoint function driver should use pci_epc_write_header() to write the standard configuration header to the endpoint controller.

- pci_epc_set_bar()

  The PCI endpoint function driver should use pci_epc_set_bar() to configure the Base Address Register in order for the host to assign PCI addr space. Register space of the function driver is usually configured using this API.

- pci_epc_clear_bar()

  The PCI endpoint function driver should use pci_epc_clear_bar() to reset the BAR.

- pci_epc_raise_irq()

  The PCI endpoint function driver should use pci_epc_raise_irq() to raise Legacy Interrupt, MSI or MSI-X Interrupt.

- pci_epc_mem_alloc_addr()

  The PCI endpoint function driver should use pci_epc_mem_alloc_addr(), to allocate memory address from EPC addr space which is required to access RC's buffer

- pci_epc_mem_free_addr()

  The PCI endpoint function driver should use pci_epc_mem_free_addr() to free the memory space allocated using pci_epc_mem_alloc_addr().

Other EPC APIs

There are other APIs provided by the EPC library. These are used for binding the EPF device with EPC device. pci-ep-cfs.c can be used as reference for using these APIs.

- pci_epc_get()

  Get a reference to the PCI endpoint controller based on the device name of the controller.

- pci_epc_put()

  Release the reference to the PCI endpoint controller obtained using pci_epc_get()

- pci_epc_add_epf()

  Add a PCI endpoint function to a PCI endpoint controller. A PCIe device can have up to 8 functions according to the specification.

- pci_epc_remove_epf()

  Remove the PCI endpoint function from PCI endpoint controller.

- pci_epc_start()

  The PCI endpoint function driver should invoke pci_epc_start() once it has configured the endpoint function and wants to start the PCI link.

- pci_epc_stop()

  The PCI endpoint function driver should invoke pci_epc_stop() to stop the PCI LINK.

PCI Endpoint Function(EPF) Library

The EPF library provides APIs to be used by the function driver and the EPC library to provide endpoint mode functionality.

EPF APIs for the PCI Endpoint Function Driver

This section lists the APIs that the PCI Endpoint core provides to be used by the PCI endpoint function driver.

- pci_epf_register_driver()

  The PCI Endpoint Function driver should implement the following ops:

  - bind: ops to perform when a EPC device has been bound to EPF device
  - unbind: ops to perform when a binding has been lost between a EPC device and EPF device
  - linkup: ops to perform when the EPC device has established a connection with a host system

  The PCI Function driver can then register the PCI EPF driver by using pci_epf_register_driver().

- pci_epf_unregister_driver()

  The PCI Function driver can unregister the PCI EPF driver by using pci_epf_unregister_driver().

- pci_epf_alloc_space()

  The PCI Function driver can allocate space for a particular BAR using pci_epf_alloc_space().

- pci_epf_free_space()

  The PCI Function driver can free the allocated space (using pci_epf_alloc_space) by invoking pci_epf_free_space().

APIs for the PCI Endpoint Controller Library

This section lists the APIs that the PCI Endpoint core provides to be used by the PCI endpoint controller library.

- pci_epf_linkup()

  The PCI endpoint controller library invokes pci_epf_linkup() when the EPC device has established the connection to the host.

Other EPF APIs

There are other APIs provided by the EPF library. These are used to notify the function driver when the EPF device is bound to the EPC device. pci-ep-cfs.c can be used as reference for using these APIs.

- pci_epf_create()

  Create a new PCI EPF device by passing the name of the PCI EPF device. This name will be used to bind the EPF device to a EPF driver.

- pci_epf_destroy()

  Destroy the created PCI EPF device.

- pci_epf_bind()

  pci_epf_bind() should be invoked when the EPF device has been bound to a EPC device.

- pci_epf_unbind()

  pci_epf_unbind() should be invoked when the binding between EPC device and EPF device is lost.