DPAA2 DPIO (Data Path I/O) Overview

This document provides an overview of the Freescale DPAA2 DPIO drivers

Introduction

A DPAA2 DPIO (Data Path I/O) is a hardware object that provides interfaces to enqueue and dequeue frames to/from network interfaces and other accelerators. A DPIO also provides hardware buffer pool management for network interfaces.

This document provides an overview the Linux DPIO driver, its subcomponents, and its APIs.

See Documentation/networking/device_drivers/ethernet/freescale/dpaa2/overview.rst for a general overview of DPAA2 and the general DPAA2 driver architecture in Linux.

Driver Overview

The DPIO driver is bound to DPIO objects discovered on the fsl-mc bus and provides services that:

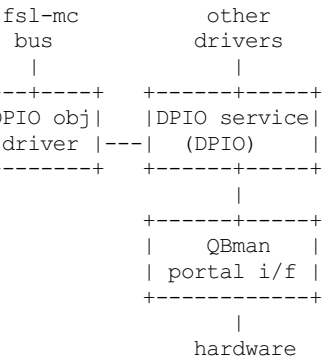
A. allow other drivers, such as the Ethernet driver, to enqueue and dequeue frames for their respective objects
B. allow drivers to register callbacks for data availability notifications when data becomes available on a queue or channel
C. allow drivers to manage hardware buffer pools

The Linux DPIO driver consists of 3 primary components--

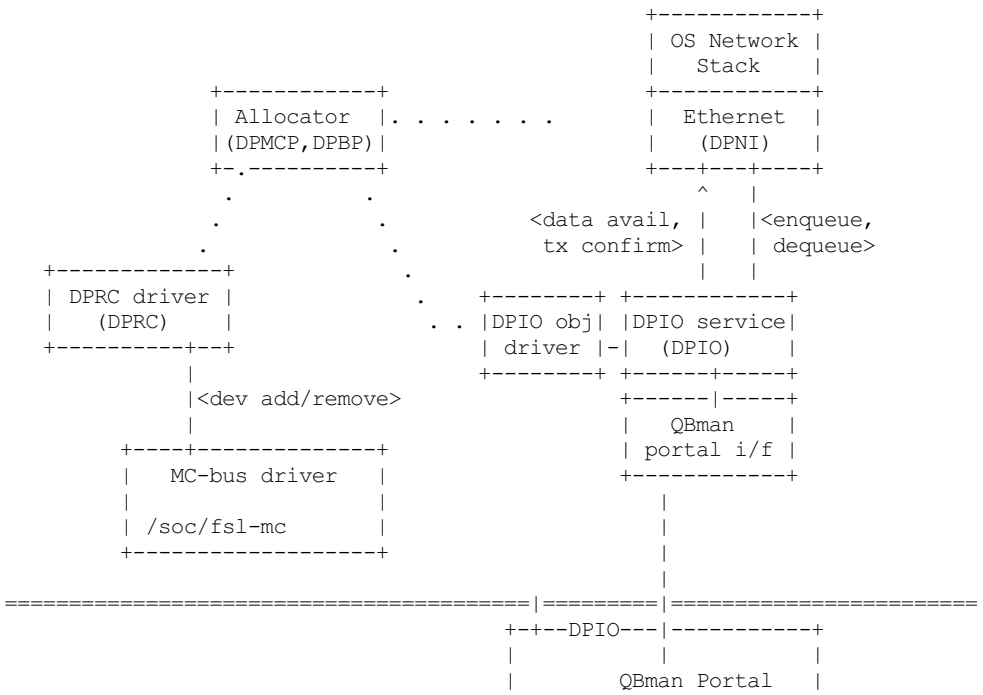
DPIO object driver-- fsl-mc driver that manages the DPIO object

DPIO service-- provides APIs to other Linux drivers for services

QBman portal interface-- sends portal commands, gets responses:



The diagram below shows how the DPIO driver components fit with the other DPAA2 Linux driver components:



+----------------------+

==========

DPIO Object Driver (dpio-driver.c)

The dpio-driver component registers with the fsl-mc bus to handle objects of type "dpio". The implementation of probe() handles basic initialization of the DPIO including mapping of the DPIO regions (the QBman SW portal) and initializing interrupts and registering irq handlers. The dpio-driver registers the probed DPIO with dpio-service.

DPIO service (dpio-service.c, dpaa2-io.h)

The dpio service component provides queuing, notification, and buffers management services to DPAA2 drivers, such as the Ethernet driver. A system will typically allocate 1 DPIO object per CPU to allow queuing operations to happen simultaneously across all CPUs.

Notification handling

dpaa2_io_service_register()

dpaa2_io_service_deregister()

dpaa2_io_service_rearm()

Queuing

dpaa2_io_service_pull_fq()

dpaa2_io_service_pull_channel()

dpaa2_io_service_enqueue_fq()

dpaa2_io_service_enqueue_qd()

dpaa2_io_store_create()

dpaa2_io_store_destroy()

dpaa2_io_store_next()

Buffer pool management

dpaa2_io_service_release()

dpaa2_io_service_acquire()

QBman portal interface (qbman-portal.c)

The qbman-portal component provides APIs to do the low level hardware bit twiddling for operations such as:

- initializing Qman software portals
- building and sending portal commands
- portal interrupt configuration and processing

The qbman-portal APIs are not public to other drivers, and are only used by dpio-service.

Other (dpaa2-fd.h, dpaa2-global.h)

Frame descriptor and scatter-gather definitions and the APIs used to manipulate them are defined in dpaa2-fd.h.

Dequeue result struct and parsing APIs are defined in dpaa2-global.h.