Texas Instruments Keystone Navigator Queue Management SubSystem driver

Driver source code path
   drivers/soc/ti/knav_qmss.c drivers/soc/ti/knav_qmss_acc.c

The QMSS (Queue Manager Sub System) found on Keystone SOCs is one of the main hardware sub system which forms the backbone of the Keystone multi-core Navigator. QMSS consist of queue managers, packed-data structure processors(PDSP), linking RAM, descriptor pools and infrastructure Packet DMA. The Queue Manager is a hardware module that is responsible for accelerating management of the packet queues. Packets are queued/de-queued by writing or reading descriptor address to a particular memory mapped location. The PDSPs perform QMSS related functions like accumulation, QoS, or event management. Linking RAM registers are used to link the descriptors which are stored in descriptor RAM. Descriptor RAM is configurable as internal or external memory. The QMSS driver manages the PDSP setups, linking RAM regions, queue pool management (allocation, push, pop and notify) and descriptor pool management.

knav qmss driver provides a set of APIs to drivers to open/close qmss queues, allocate descriptor pools, map the descriptors, push/pop to queues etc. For details of the available APIs, please refers to include/linux/soc/ti/knav_qmss.h

DT documentation is available at Documentation/devicetree/bindings/soc/ti/keystone-navigator-qmss.txt

Accumulator QMSS queues using PDSP firmware

The QMSS PDSP firmware support accumulator channel that can monitor a single queue or multiple contiguous queues. drivers/soc/ti/knav_qmss_acc.c is the driver that interface with the accumulator PDSP. This configures accumulator channels defined in DTS (example in DT documentation) to monitor 1 or 32 queues per channel. More description on the firmware is available in CPPI/QMSS Low Level Driver document (docs/CPPI_QMSS_LLD_SDS.pdf) at

   git://git.ti.com/keystone-rtos/qmss-lld.git

k2_qmss_pdsp_acc48_k2_le_1_0_0_9.bin firmware supports upto 48 accumulator channels. This firmware is available under ti-keystone folder of firmware.git at

   git://git.kernel.org/pub/scm/linux/kernel/git/firmware/linux-firmware.git

To use copy the firmware image to lib/firmware folder of the initramfs or ubifs file system and provide a sym link to k2_qmss_pdsp_acc48_k2_le_1_0_0_9.bin in the file system and boot up the kernel. User would see

   "firmware file ks2_qmss_pdsp_acc48.bin downloaded for PDSP"

in the boot up log if loading of firmware to PDSP is successful.

Use of accumulated queues requires the firmware image to be present in the file system. The driver doesn't acc queues to the supported queue range if PDSP is not running in the SoC. The API call fails if there is a queue open request to an acc queue and PDSP is not running. So make sure to copy firmware to file system before using these queue types.