RapidIO subsystem mport character device driver (rio_mport_cdev.c)

1. Overview

This device driver is the result of collaboration within the RapidIO.org Software Task Group (STG) between Texas Instruments, Freescale, Prodrive Technologies, Nokia Networks, BAE and IDT. Additional input was received from other members of RapidIO.org. The objective was to create a character mode driver interface which exposes the capabilities of RapidIO devices directly to applications, in a manner that allows the numerous and varied RapidIO implementations to interoperate.

This driver (MPORT_CDEV) provides access to basic RapidIO subsystem operations for user-space applications. Most of RapidIO operations are supported through 'ioctl' system calls.

When loaded this device driver creates filesystem nodes named rio_mportX in /dev directory for each registered RapidIO mport device. 'X' in the node name matches to unique port ID assigned to each local mport device.

Using available set of ioctl commands user-space applications can perform following RapidIO bus and subsystem operations:

- Reads and writes from/to configuration registers of mport devices (RIO_MPORT_MAINT_READ_LOCAL/RIO_MPORT_MAINT_WRITE_LOCAL)
- Reads and writes from/to configuration registers of remote RapidIO devices. This operations are defined as RapidIO Maintenance reads/writes in RIO spec. (RIO_MPORT_MAINT_READ_REMOTE/RIO_MPORT_MAINT_WRITE_REMOTE)
- Set RapidIO Destination ID for mport devices (RIO_MPORT_MAINT_HDID_SET)
- Set RapidIO Component Tag for mport devices (RIO_MPORT_MAINT_COMPTAG_SET)
- Query logical index of mport devices (RIO_MPORT_MAINT_PORT_IDX_GET)
- Query capabilities and RapidIO link configuration of mport devices (RIO_MPORT_GET_PROPERTIES)
- Enable/Disable reporting of RapidIO doorbell events to user-space applications (RIO_ENABLE_DOORBELL_RANGE/RIO_DISABLE_DOORBELL_RANGE)
- Enable/Disable reporting of RIO port-write events to user-space applications (RIO_ENABLE_PORTWRITE_RANGE/RIO_DISABLE_PORTWRITE_RANGE)
- Query/Control type of events reported through this driver: doorbells, port-writes or both (RIO_SET_EVENT_MASK/RIO_GET_EVENT_MASK)
- Configure/Map mport's outbound requests window(s) for specific size, RapidIO destination ID, hopcount and request type (RIO_MAP_OUTBOUND/RIO_UNMAP_OUTBOUND)
- Configure/Map mport's inbound requests window(s) for specific size, RapidIO base address and local memory base address (RIO_MAP_INBOUND/RIO_UNMAP_INBOUND)
- Allocate/Free contiguous DMA coherent memory buffer for DMA data transfers to/from remote RapidIO devices (RIO_ALLOC_DMA/RIO_FREE_DMA)
- Initiate DMA data transfers to/from remote RapidIO devices (RIO_TRANSFER). Supports blocking, asynchronous and posted (a.k.a 'fire-and-forget') data transfer modes.
- Check/Wait for completion of asynchronous DMA data transfer (RIO_WAIT_FOR_ASYNC)
- Manage device objects supported by RapidIO subsystem (RIO_DEV_ADD/RIO_DEV_DEL). This allows implementation of various RapidIO fabric enumeration algorithms as user-space applications while using remaining functionality provided by kernel RapidIO subsystem.

2. Hardware Compatibility

This device driver uses standard interfaces defined by kernel RapidIO subsystem and therefore it can be used with any mport device driver registered by RapidIO subsystem with limitations set by available mport implementation.

At this moment the most common limitation is availability of RapidIO-specific DMA engine framework for specific mport device. Users should verify available functionality of their platform when planning to use this driver:

- IDT Tsi721 PCIe-to-RapidIO bridge device and its mport device driver are fully compatible with this driver.
- Freescale SoCs 'fsl_rio' mport driver does not have implementation for RapidIO specific DMA engine support and therefore DMA data transfers mport_cdev driver are not available.

3. Module parameters

- 'dma_timeout'
  - DMA transfer completion timeout (in msec, default value 3000). This parameter set a maximum completion wait time for SYNC mode DMA transfer requests and for RIO_WAIT_FOR_ASYNC ioctl requests.
- 'dbg_level'
  - This parameter allows to control amount of debug information generated by this device driver. This parameter is

formed by set of bit masks that correspond to the specific functional blocks. For mask definitions see 'drivers/rapidio/devices/rio_mport_cdev.c' This parameter can be changed dynamically. Use CONFIG_RAPIDIO_DEBUG=y to enable debug output at the top level.

4. Known problems

None.

5. User-space Applications and API

API library and applications that use this device driver are available from RapidIO.org.

6. TODO List

- Add support for sending/receiving "raw" RapidIO messaging packets.
- Add memory mapped DMA data transfers as an option when RapidIO-specific DMA is not available.