Linux Switchtec Support

Microsemi's "Switchtec" line of PCI switch devices is already supported by the kernel with standard PCI switch drivers. However, the Switchtec device advertises a special management endpoint which enables some additional functionality. This includes:

- Packet and Byte Counters
- Firmware Upgrades
- Event and Error logs
- Querying port link status
- Custom user firmware commands

The switchtec kernel module implements this functionality.

Interface

The primary means of communicating with the Switchtec management firmware is through the Memory-mapped Remote Procedure Call (MRPC) interface. Commands are submitted to the interface with a 4-byte command identifier and up to 1KB of command specific data. The firmware will respond with a 4-byte return code and up to 1KB of command-specific data. The interface only processes a single command at a time.

Userspace Interface

The MRPC interface will be exposed to userspace through a simple char device: /dev/switchtec#, one for each management endpoint in the system.

The char device has the following semantics:

- A write must consist of at least 4 bytes and no more than 1028 bytes. The first 4 bytes will be interpreted as the Command ID and the remainder will be used as the input data. A write will send the command to the firmware to begin processing.
- Each write must be followed by exactly one read. Any double write will produce an error and any read that doesn't follow a write will produce an error.
- A read will block until the firmware completes the command and return the 4-byte Command Return Value plus up to 1024 bytes of output data. (The length will be specified by the size parameter of the read call -- reading less than 4 bytes will produce an error.)
- The poll call will also be supported for userspace applications that need to do other things while waiting for the command to complete.

The following IOCTLs are also supported by the device:

- SWITCHTEC_IOCTL_FLASH_INFO - Retrieve firmware length and number of partitions in the device.
- SWITCHTEC_IOCTL_FLASH_PART_INFO - Retrieve address and lengeth for any specified partition in flash.
- SWITCHTEC_IOCTL_EVENT_SUMMARY - Read a structure of bitmaps indicating all uncleared events.
- SWITCHTEC_IOCTL_EVENT_CTL - Get the current count, clear and set flags for any event. This ioctl takes in a switchtec_ioctl_event_ctl struct with the event_id, index and flags set (index being the partition or PFF number for non-global events). It returns whether the event has occurred, the number of times and any event specific data. The flags can be used to clear the count or enable and disable actions to happen when the event occurs. By using the SWITCHTEC_IOCTL_EVENT_FLAG_EN_POLL flag, you can set an event to trigger a poll command to return with POLLPRI. In this way, userspace can wait for events to occur.
- SWITCHTEC_IOCTL_PFF_TO_PORT and SWITCHTEC_IOCTL_PORT_TO_PFF convert between PCI Function Framework number (used by the event system) and Switchtec Logic Port ID and Partition number (which is more user friendly).

Non-Transparent Bridge (NTB) Driver

An NTB hardware driver is provided for the Switchtec hardware in ntb_hw_switchtec. Currently, it only supports switches configured with exactly 2 NT partitions and zero or more non-NT partitions. It also requires the following configuration settings:

- Both NT partitions must be able to access each other's GAS spaces. Thus, the bits in the GAS Access Vector under Management Settings must be set to support this.
- Kernel configuration MUST include support for NTB (CONFIG_NTB needs to be set)

NT EP BAR 2 will be dynamically configured as a Direct Window, and the configuration file does not need to configure it explicitly.

Please refer to Documentation/driver-api/ntb.rst in Linux source tree for an overall understanding of the Linux NTB stack. ntb_hw_switchtec works as an NTB Hardware Driver in this stack.