NAME

mlx5io - IOCTL interface to manage Connect-X 4/5/6 Mellanox network adapters

SYNOPSIS

#include <dev/mlx5/mlx5io.h>

DESCRIPTION

The **mlx5io** interface is provided for management of the Connect-X4, 5 and 6 network adapters in the aspects not covered by the generic network configuration, mostly related to the PCIe attachment and internal card working. Interface consists of the commands, which are passed by means of ioctl(2) on the file descriptor, opened from the */dev/mlx5ctl* device node.

The following commands are implemented:

MLX5_FWDUMP_FORCE

Take the snapshot of the firmware registers state and store it in the kernel buffer. The buffer must be empty, in other words, no dumps should be written so far, or existing dump cleared with the MLX5_FWDUMP_RESET command for the specified device. The argument for the command should point to the *struct mlx5_tool_addr* structure, containing the PCIe bus address of the device.

```
struct mlx5_tool_addr {
        uint32_t domain;
        uint8_t bus;
        uint8_t slot;
        uint8_t func;
};
```

MLX5_FWDUMP_RESET

Clear the stored firmware dump, preparing the kernel buffer for the next dump. The argument for the command should point to the *struct mlx5_tool_addr* structure, containing the PCIe bus address of the device.

MLX5_FWDUMP_GET

Fetch the stored firmware dump into the user memory. The argument to the command should point to the input/output *struct mlx5_fwdump_get* structure. Its devaddr field specifies the address of the device, the buf fields points to the array of *struct mlx5_fwdump_reg* of records of the registers values, the size of the array is specified in the reg_cnt field.

```
struct mlx5_fwdump_get {
```

```
        struct mlx5_tool_addr devaddr;
        struct mlx5_fwdump_reg *buf;
        size_t reg_cnt;
        size_t reg_filled; /* out */
};
```

On successfull return, the reg_filled field reports the number of the buf array elements actually filled with the registers values. If buf contains the NULL pointer, no registers are filled, but reg_filled still contains the number of registers that should be passed for the complete dump.

The *struct mlx5_fwdump_reg* element contains the address of the register in the field addr, and its value in the field val.

```
struct mlx5_fwdump_reg {
        uint32_t addr;
        uint32_t val;
};
```

MLX5_FW_UPDATE

Requests firmware update (flash) on the adapter specified by the devaddr using the firmware image in MFA2 format. The argument for the ioctl command is the *struct mlx5_fw_update* with the following definition.

```
struct mlx5_fw_update {
        struct mlx5_tool_addr devaddr;
        void *img_fw_data;
        size_t img_fw_data_len;
};
```

Image address in memory is passed in img_fw_data, the length of the image is specified in img_fw_data_len field.

MLX5_FW_RESET

Requests PCIe link-level reset on the device. The address of the device is specified by the *struct mlx5_tool_addr* structure, which should be passed as an argument.

MLX5_EEPROM_GET

Fetch EEPROM information. The argument to the command should point to the input/output *struct mlx5_eeprom_get* structure where, the devaddr field specifies the address of the device.

```
struct mlx5_eeprom_get {
```

```
        struct mlx5_tool_addr devaddr;
        size_t eeprom_info_page_valid;
        uint32_t *eeprom_info_buf;
        size_t eeprom_info_out_len;
};
```

On successfull return, the eeprom_info_out_len field reports the length of the EEPROM information. eeprom_info_buf field contains the actual EEPROM information. eeprom_info_page_valid field reports the third page validity.

FILES

The */dev/mlx5ctl* devfs(5) node is used to pass commands to the driver.

RETURN VALUES

If successful, the IOCTL returns zero. Otherwise, -1 is returned and the global variable *errno* is set to indicate the error.

SEE ALSO

errno(2), ioctl(2), mlx5en(4), mlx5ib(4), mlx5tool(8) and pci(9).